

高等学校教材

电子技术综合 实验指导书

华北电力大学电子技术课程组 编

主 编 孙淑艳

副主编 赵 东 柳 赞

高等教育出版社

高等学校教材

电子技术综合 实验指导书

Dianzi Jishu Zonghe Shiyān Zhidāoshu

华北电力大学电子技术课程组 编

主 编 孙淑艳

副主编 赵 东 柳 赞

高等教育出版社·北京

内容提要

本书是电子技术基础课程系列教材之一,是根据教育部高等学校电子电气基础课程教学指导分委会最新制定的电子技术课程教学基本要求编写的一本适用面较宽、实用性较强的电子技术综合设计的教材。

全书共包括八个部分七个章节。主要内容包括电子技术综合设计概述、NI Multisim 12 使用指南、电子技术综合实验 Multisim 仿真部分、VHDL 语言简介、电子技术综合实验 Quartus II 部分、手工焊接技术简介、电子技术综合实验硬件电路设计部分以及附录。其中,电子技术综合实验 Multisim 仿真部分包括 12 个题目,电子技术综合实验 Quartus II 部分包括 6 个题目,电子技术综合实验硬件电路设计部分包括 3 个题目,附录中包括集成电路简介,集成逻辑电路的连接和驱动,集成逻辑门电路新、旧图形符号对照,集成触发器新、旧图形符号对照,常用数字集成电路型号及引脚图以及综合实验的设计报告等供读者参考。

本书可作为高等院校电气工程及其自动化、电子信息工程、电子科学与技术、通信工程、自动化、测控技术与仪器、计算机科学与技术、软件工程等专业的本、专科教材,同时也可作为参加各类电子设计竞赛学生自学的参考书,以及相关工程技术人员的参考书。

图书在版编目(CIP)数据

电子技术综合实验指导书. 孙耀锋, 华耀锋, 张泽业, 刘莉, 张泽业编. —北京: 高等教育出版社, 2015.9

ISBN 978-7-04-043200-9

I. ①电… II. ①孙… ②华… III. ①电子技术-实验-高等学校-教材 IV. ①TN-33

中国版本图书馆 CIP 数据核字(2015)第 171826 号

策划编辑 王耀锋 责任编辑 王耀锋 封面设计 张 志 版式设计 马敬茹
插图绘制 杜晓丹 责任校对 刘 莉 责任印制 张泽业

出版发行 高等教育出版社
社 址 北京市西城区德外大街 4 号
邮政编码 100120
印 刷 三河市潮河印业有限公司
开 本 787mm×1092mm 1/16
印 张 15
字 数 340 千字
购书热线 010-58581118

咨询电话 400-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>
网上订购 <http://www.landracom.com>
<http://www.landracom.com.cn>
版 次 2015 年 9 月第 1 版
印 次 2015 年 9 月第 1 次印刷
定 价 22.30 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换
版权所有 侵权必究
物 料 号 43290-00

前 言

为贯彻落实教育部“关于进一步加强高等学校本科教学工作的若干意见”文件的精神,根据当前教育教学改革的发展趋势,本着满足当代大学生的知识结构、综合能力、创新能力等方面的需求,华北电力大学电子技术课程组编写了一套电子技术基础课程系列教材。该系列教材是编者对高等学校电工电子系列课程的内容和体系进行了深入研究,针对华北电力大学电类和非电类专业的特点,并总结多年理论教学和实践经验的基础上编写而成的。本套系列教材充分体现了工程技术教育的特点,力求达到教学与实验相结合、理论与应用相统一,培养学生运用电子技术解决实际问题的工程能力和实践能力。其主要内容覆盖了模拟电子技术、数字电子技术、电子电路的测试技术以及计算机辅助分析和设计方法等。本套教材包括《电子技术基础》、《模拟电子技术基础》、《数字电子技术基础》、《电子技术实验指导书》、《模拟电子技术实验指导书》、《数字电子技术实验指导书》、《电子技术综合实验指导书》。

本书是该系列教材的一部。

电子技术综合实验是电气信息类及其他相近专业学生在完成模拟电子技术基础和数字电子技术基础这两门课程学习之后集中进行的实验、实践环节。电子技术综合实验全部为综合性、设计性实验,需要学生综合运用所学电子学知识进行分析设计,对学生工程实践能力和科研创新能力的培养很有帮助,是实践教学中不可缺少的环节。《电子技术综合实验指导书》包含八个部分,主要内容有电子技术综合设计概述,NI Multisim 12 使用指南,电子技术综合实验 Multisim 部分,VHDL 语言简介,电子技术综合实验 Quartus II,手工焊接技术简介,电子技术综合实验硬件电路设计部分和附录。其中,电子技术综合实验 Multisim 仿真部分包含 12 个题目,电子技术综合实验 Quartus II 部分包含 6 个题目,电子技术综合实验硬件电路设计部分包含 3 个题目,附录中介绍集成电路简介,集成逻辑电路的连接和驱动,集成逻辑门电路新、旧图形符号对照,集成触发器新、旧图形符号对照,常用数字集成电路型号及引脚图以及综合实验的设计报告等。

本实验指导书主要有以下特点。

1. 软件仿真与硬件电路连接相结合,使学生能够完整地了解、学习电子设计的全过程。
2. 辅助理论教学,循序渐进。通过仿真实验,对 Multisim 和 Quartus II 两款软件的学习,弥补了理论教学上机学时不足的缺憾。
3. 设计题目典型,实用性强。本书设计题目都是典型的 EDA 设计项目,注重电子设计的实用性,有助于学生加深对电子电路设计的认识,有助于提高学生的动手能力和创新能力。
4. 启发创新,促进协作。本书设计了多种题目,学生可以有很大的选题空间,通过自行组队、自行设计、自行安装和调试,完成具有一定创意的选题,在题目实现过程中可以培养学生的创

II 前言

新意识,增强团队的协作精神。

参加本书编写工作的有孙淑艳(第一章、第二章、第三章、第六章和附录 A、B、F),赵东(第四章、第五章),柳赞(第二章、第七章和附录 C、D、E)。由孙淑艳任主编,赵东、柳赞任副主编,戴振刚老师担任本书的主审。本书的编写得到华北电力大学电气与电子工程学院张瑞华、张根保、张青等多位退休老教师的帮助,在此,向他们表示感谢。

我们对在本书编写和出版过程中给予热情帮助和支持的同志们表示深切的谢意。

本书尚有许多待改进之处,敬请读者在使用本书时,将发现的问题及时指出,并将意见和建议及时反馈给我们,以便今后不断改进。编者邮箱为 sshy@ncepu.edu.cn。

编 者

2015年6月于华北电力大学

目 录

第一章 电子技术综合设计概述	1	4.1 VHDL 程序基础	114
1.1 电子技术综合设计课程	1	4.1.1 概述	114
1.2 电子技术综合设计过程	2	4.1.2 程序实体(ENTITY)	116
1.3 EDA 技术简介	6	4.1.3 结构体(ARCHITECTURE)	118
1.4 数字系统概述	7	4.2 VHDL 词法基础	119
1.5 数字系统设计实例	9	4.2.1 标识符	119
第二章 NI Multisim 12 使用指南	16	4.2.2 VHDL 数据对象	120
2.1 NI Multisim 12 简介	16	4.2.3 VHDL 数据类型	121
2.2 NI Multisim 12 的基本操作界面	18	4.2.4 VHDL 操作符	124
2.3 元器件库	23	4.2.5 注释与数字表示	128
2.4 仪器仪表库	29	4.2.6 预定义属性	129
2.5 Multisim 12 的基本操作	69	4.3 VHDL 顺序语句结构	132
第三章 电子技术综合实验 Multisim		4.3.1 赋值语句	132
仿真部分	86	4.3.2 流程控制语句	133
题目一 移位寄存器型彩灯控制器	86	4.3.3 WAIT 语句	137
题目二 智力竞赛抢答器	90	4.3.4 NULL 语句	138
题目三 电子拔河游戏机	91	4.3.5 RETURN 语句	138
题目四 交通信号灯控制器	93	4.3.6 过程调用语句	138
题目五 电子秒表	95	4.4 VHDL 并行语句结构	138
题目六 电子密码锁	97	4.4.1 进程语句	139
题目七 数字电子钟	98	4.4.2 块语句	141
题目八 电子爆竹的设计	100	4.4.3 并行信号赋值语句	142
题目九 数字频率计	101	4.4.4 过程调用语句	145
题目十 用运算放大器组成万用电表		4.4.5 元件例化语句	146
的设计与调试	103	4.4.6 类属映射语句	147
题目十一 温度检测及控制电路	106	4.4.7 生成语句	147
题目十二 随机存取存储器 2114A		4.4.8 库 LIBRARY	148
及其应用	110	4.4.9 程序包 PACKAGE	150
第四章 VHDL 语言简介	114	4.4.10 配置 CONFIGURATION	151

II 目录

4.5 Quartus II 使用指南	152	6.4 拆焊	197
4.5.1 Quartus II 开发工具的介绍	152	第七章 电子技术综合实验硬件电路设计	
4.5.2 文本设计输入	152	部分	199
4.5.3 设计文件编译	158	7.1 面包板及其使用	199
4.5.4 Quartus II 开发工具的仿真分析	162	7.2 数字电子钟的设计、安装与	
4.5.5 器件编程	168	调试	202
第五章 电子技术综合实验 Quartus II		7.3 智力竞赛抢答器设计、安装与	
部分	179	调试	207
题目一 3-8 译码器的设计	179	7.4 电子秒表的设计、安装与调试	209
题目二 七段数码管显示	181	附录 A 集成电路简介	211
题目三 四位串行全加器的设计	184	附录 B 集成逻辑电路的连接和驱动	215
题目四 六十进制计数器的设计	186	附录 C 集成逻辑门电路新、旧图形	
题目五 可编程彩灯控制器的设计	187	符号对照	218
题目六 乒乓球游戏机的设计与		附录 D 集成触发器新、旧图形符号	
实现	188	对照	220
第六章 手工焊接技术简介	189	附录 E 常用数字集成电路型号及	
6.1 锡焊原理	189	引脚图	222
6.2 焊接工具和材料	190	附录 F 综合实验的设计报告	228
6.3 手工焊接的基本技能	195	参考文献	233

第一章

电子技术综合设计概述

电子技术综合设计是电气信息类及其他相近专业的学生在完成模拟电子技术基础和数字电子技术基础这两门课程学习之后集中进行的实验、实践环节。电子技术综合设计题目全部为综合性、设计性的题目,学生需在教师指导下,独立完成某一题目的系统设计、安装和调试工作的全过程。通过电子技术综合设计培养学生具备一定专业技术能力和综合运用所学知识的能力,使之具有理论联系实际、独立分析和解决实际问题的能力。

1.1 电子技术综合设计课程

一、电子技术综合设计的基本要求

(1) 能够根据设计任务和指标要求,综合运用电子技术课程中所学到的理论知识与实践技能独立完成一个设计课题。

(2) 根据课题需要选择参考书籍,查阅手册、图表等有关文献资料。要求通过独立思考、深入钻研课程设计中遇到的问题,培养自行分析、解决问题的能力。

(3) 进一步熟悉常用电子器件的类型和特性,掌握合理选用元器件的原则。

(4) 学会电子电路的安装与调试技能,掌握常用仪器仪表的正确使用方法。利用“观察、判断、实验、再判断”的基本方法,解决实验中出现的的问题。

(5) 学会撰写课程设计总结报告。

(6) 通过课程设计,逐步形成严肃认真、一丝不苟、实事求是的工作作风和科学态度,培养学生树立一定的生产观点、经济观点和全局观点。要求学生在设计过程中,坚持勤俭节约的原则,从现有条件出发,将损坏元器件的比率降到最小。

(7) 在课程设计过程中,要做到爱护公物、遵守纪律、团结协作、注意安全。

二、电子技术综合设计的一般步骤

电子技术综合设计是指在教师的指导下,学生依据事先提供的技术指标和功能,综合运用电子技术平台所提供的知识独立完成某一设计题目,达到综合训练的目的。一般在确定设计题目后,就可以按照以下几个阶段进行设计。

(1) 整体规划:从设计题目的任务、要求和条件出发,将整体设计分解为多个单元模块或者多个步骤,然后通过论证与选择,确定总体设计方案。

(2) 具体设计和调试:根据不同单元模块或不同步骤的功能,通过计算,查找可用器件,并辅以外围电路进行计算机辅助分析与设计,当各项指标满足设计要求时,搭接具体的硬件单元电路,再对硬件单元电路进行调试,如果满足不了设计要求和指标,则需要进行修改,直到调试结果满足了单元模块功能的设计要求和指标。

(3) 整合调试:将调试好的单元模块进行组合或者将不同步骤按顺序连接成系统,对整个系统按设计要求和指标进行分项调试,当满足了设计要求时,可以撰写设计报告,否则要进行局部修改,甚至推翻之前的所有方案和设计重新进行设计,直到系统指标达到设计要求。

(4) 撰写报告:设计报告是学生电子技术设计全过程的系统总结,是对学生归纳技术资料、撰写报告以及对参数分析能力的训练。设计报告应该包括封面、目录、实验任务书、正文、附录和参考文献等几个部分。同时设计报告字迹要清楚、文理要通顺,数据要有理论分析、要实事求是,表格和电路图要用尺子画,曲线用坐标纸描绘。

1.2 电子技术综合设计过程

一个完整的电子系统的设计和制作是比较复杂的,一般包括电子电路系统的硬件电路设计,仿真分析,印刷电路设计,安装、调试和测试系统指标四个过程。下面分别介绍这四个过程的具体操作。

一、电子电路系统的硬件电路设计

1. 单元电路设计

首先,结合现有的条件将系统的原理方案框图具体化。系统原理框图只考虑从原理上如何实现任务要求,其中每一部分只是原理功能的描述,其实现方法是多种多样的。因此,要结合设计者的知识储备和现有的器件条件,将原理框图具体化。也就是说,确定用什么器件(通用集成电路、专用集成电路、PLD、单片机和 ARM 等)和电路(各种运算电路、比较电路、波形发生电路、编码器、译码器、数据选择器、计数器、寄存器、存储器、A/D、D/A 电路等)实现原理框图中的各

部分功能,为设计单元电路做准备。接着进行单元电路设计。根据方案要求,将任务要求和技术指标等进行分解和分配,明确对各单元电路的要求,详细拟订各单元电路的性能指标,并注意各单元电路的输入信号、输出信号和控制信号之间的关系与相互配合。电路设计的工作主要是选择或设计单元电路的结构形式,确定元器件的参数及型号等。在选择单元电路的结构形式时,首选的方法是从已学过的知识和文献资料中寻找合适的电路形式,修改部分元器件参数,这一般能满足要求。需要自己设计的电路一般是各功能单元之间的接口电路、保证集成电路正常工作的外围电路和不适合集成电路完成的功能电路(如高电压、大电流、大功率电路等)。有了电路的拓扑结构,就可以根据电路理论计算各元器件的参数,设计时,一般要选用 EDA(如 Multisim 仿真软件)工具对电路性能进行仿真分析。确定元器件参数后,经优化选择分析,选择元器件的型号。要注意的是,选择的元器件要留有一定的余量(通常要大于额定值的 1.5 倍),按电路的最坏条件选择。满足指标要求的电路、元器件等一般不只一个,应该进行分析比较,从性能、价格、体积、货源、器件间参数配合是否合理等方面综合考虑、择优选用。各种单元电路的参考资料很多,有的可直接借用,有的略加改进即可应用,有的设计思想有启发性,对设计者有参考价值。

2. 总体电路原理图设计

总体电路原理图是准备元器件和进行 PCB(印刷电路板)设计、电路安装和调试的依据,是最重要的设计文件之一,必须正确、规范。画总体电路原理图有助于找出设计中存在的问题,尤其是各单元之间的连接问题。画原理图时要注意合理布局,排列均匀,横平竖直,连线清晰。原理图设计完成之后要进行一次认真、全面的审核。因为在设计过程中有些问题难免考虑不周,甚至出错,且在设计时对各单元电路实现的功能考虑较多,前后电路的配合可能放在次要的位置。审核的主要内容如下:① 级与级之间的电气性能是否匹配,如阻抗、线性范围、负载能力和电平;② 各信号之间的时序是否能配合,尤其对时序电路,各信号之间有确定的时序关系;如时序配合不符,系统将无法正常工作;③ 信号耦合方式是否合理,如采用阻容耦合、直接耦合、光电耦合和变压器耦合等。

二、电子电路系统的仿真分析

对于复杂的电子电路系统,为提高设计效率,通常在设计 PCB(印刷电路板)之前要进行仿真分析。通过仿真分析,一方面可验证设计原理的可行性,另一方面可以通过修改电路的结构和元器件参数使得设计的电路更加合理。仿真分析时,先分析单元电路,待单元电路的各项指标符合设计要求时,再对整体电路进行仿真分析。一般都要经过仿真、观察结果和调整元件参数的反复过程。

1. 单元电路的仿真分析

在进行单元电路分析时,可按照一定的规则(如对于数据采集系统可按照信号的流向分为输入模块、信号处理模块和输出模块)将系统分成若干模块单元,分别进行单元电路的仿真分析,待全部单元电路信号仿真结束以后,将单元电路进行封装,封装时要注意各模块单元之间的信号连接并加以适当地注解,防止电路结构复杂以后,在总体电路设计时,模块之间的连线产生混淆。

2. 总体电路的仿真分析

在进行总体电路仿真时,将封装好的单元模块连接成整体电路,再加上电源、信号源以及显示模块。按设计要求,设定电源的数值,信号源的幅值、频率、相位等参数,启动仿真分析,观察输出结果,若不符合设计要求,通过调整电源和信号源参数,看输出结果是否满足要求,有时要通过调整单元模块电路的结构和参数,才能达到设计要求。在电路仿真时,改变电路的结构和元器件参数都非常方便。

3. 仿真时应注意的问题

(1) 为提高仿真分析的效率,对于有的 EDA 软件(如 Multisim)仿真时,首先选用虚拟元器件模型创建电路,然后进行仿真分析。因为虚拟元器件的参数调整方便,使设计的电路很容易满足设计要求,然后再用实际元器件模型创建电路,最后将两者仿真分析的结果进行比较,以便设计出合理的电路结构和元器件参数。

(2) 在进行仿真分析时,要考虑最坏情况(如温度的变化对放大电路性能的影响,信号频率的改变对放大倍数的影响,数字电路的竞争冒险现象等)来选择电路结构和元器件,并观察和分析仿真结果。要充分利用仿真软件提供的虚拟仪器、各种仿真分析方法以及元器件的故障诊断工具,对设计的电路做充分的仿真分析,并和理论分析的结果相比较,若存在较大差异,要设法找出原因,及时加以解决,不遗留问题。因为仿真分析是对理论分析的验证,是实际设计的指导依据。

三、电子电路系统的印刷电路设计

对于电子电路系统来说,印刷电路板设计是不可缺少的环节。产品质量的好坏,除了与元器件质量及电路结构有关外,印刷电路板设计是否合理也是一个关键因素。随着电子技术的发展,印刷电路板上的元器件布局和走线的密度越来越高,对印刷电路板设计的要求也越来越高。因此,在进行印刷电路板设计时必须遵守设计的一般原则,考虑散热、电磁兼容性、抗干扰等因素,合理布局,正确走线,便于生产中的安装、调试与检修。设计时应注意以下问题。

(1) 印刷电路板的尺寸要合适。除了受外壳大小限制,它也受其他因素影响,如尺寸过大时,印刷导线长,阻抗增加,抗噪声能力下降,成本增加;尺寸过小则元件布局过密,散热性能不好,且相邻的线条易受干扰。

(2) 元器件布局的一般原则。元器件应尽量均匀、整齐、紧凑地排列在印刷电路板上。每个元器件引出端应有单独的焊盘,不允许两个元器件引出端共用一个焊盘。元器件不允许交叉和重叠安装。应尽量使元器件平行排列。

(3) 特殊元器件布局。印刷电路板与外界的连接器件应安装在边缘适当位置上,以便于与外界连线的走线,相关联的引出线端不要距离太远,且进出线端尽可能集中在 1 至 2 个侧面,不要过于分散。

(4) 布线设计基本原则。单面板中不允许有交叉电路,对于可能交叉的线条,可以用“钻”(即引线从元件引脚间的空隙处“钻”过去)、“绕”(即从可能交叉的引线的一端“绕”过去)两种办法解决。有些线路实在无法连接时,可以采用跨接线来连接,但应尽量少用。印刷导线要尽可

能短,导线的折弯应成圆角以免影响电气性能。双面板布线时,两面导线宜相互垂直、斜交和弯曲走线,避免相互平行,以减小寄生耦合。电路输入及输出用的印刷导线应尽量避免相互平行,这些导线间最好加接地线。

四、安装、调试和测试系统指标

1. 安装电路

当电子电路系统比较复杂时,系统电路使用的集成电路和元器件多,连线复杂,在安装电路前,应检查元器件,包括型号、种类、数量等,对关键的元器件要进行测试验证;在安装元器件时,要注意器件的引脚排列,不能接错,否则会使器件损坏,并增加排除故障的难度,浪费时间。元器件在安装完毕后,一般不要急于通电调试,应从安装质量角度进行外观检查,从人身安全和设备安全角度进行安全检查。外观检查主要检查元器件的安装情况,如元器件的型号规格与电路中标出的是否一致,电解电容的极性是否接错等。

2. 电路的调试和测试

电路的调试按模块进行,若电路中含有电源模块,首先要调试电源模块,看电源模块的工作是否正常,包括电源的极性、电源的幅值等是否符合要求,在电源模块正常的情况下,再调试其他模块。其他模块的调试顺序一般按信号的流向逐个进行,逐步扩大调试范围,将前面调试过的模块输出信号作为后一级模块的输入信号,为最后的整体调试创造条件。

电子电路系统的调试步骤大致如下。

(1) 通电观察。印刷电路板安装完元器件后,在确认电路接上电源后不会有异常的大电流产生后,再将电源接入电路。通电后不要急于测量电气指标,而要观察电路有无异常现象,如有无冒烟现象、有无异常气味;手摸集成电路外封装,是否发热、发烫等。如果出现异常现象,应立即关掉电源,待故障排除后再通电。电路通电后,以电源地为基准,测量各路电源电压和主要元器件的电源电压,以确保各元器件在正常电源电压下工作,同时测量各器件地引脚的电位,保证电位为零。

(2) 静态调试。静态调试一般是指在不加输入信号(或只加固定的电平信号)的条件下进行的直流测试和直流工作点的调整过程。可用万用表测量电路中各点的电位,通过和理论估算值进行比较,结合电路原理的分析以及仿真结果,判断电路的直流工作状态是否正常,及时发现电路中已损坏或处于临界状态的元器件。通过更换器件或调整电路参数,使电路的直流工作状态满足要求。

(3) 动态调试。动态调试是在静态调试的基础上进行的,在电路的输入端加入合适的信号,按信号的流向,顺序检测各测试点的输出信号,若发现不正常现象,应分析其原因并排除故障,再进行调试,直到满足要求。

(4) 分块调试完成后,接好各功能模块之间的接口电路,就可进行总体电路的总调。总调主要对电路的性能指标进行测试和调整,若不符合设计要求,应仔细分析其原因,找出相应的单元进行调整。对有把握不需调试的电路或不适合分块调试的电路,可直接进行总体调试。电路调试时经常遇到电路通电后没有正常输出的情况,这时不要紧张,要冷静分析(如接触不良、元器

件参数不合理、原理图本身有误等),尽快找到故障原因。为了能尽快找到故障位置,必须熟悉整个电路、各功能模块和有关单元电路的结构、工作原理和器件特性,清楚电路中各测试点测量值的正常范围。通过比较实测值和正常值,可找出故障的大致位置。总的思路是通过有限的观察、测量和分析,尽快将故障缩小到某一功能单元,使故障原因的分析、查找变得简单。常用的查找故障的步骤如下。

① 检查供电系统。特别是在电路不工作、测量不到测试点电位时,除了查电路系统的电源进线外,对关键元器件及集成电路的电源和地也要进行测量,以确保电路中主要元器件在正常的电源电压下工作。

② 确定故障模块。将总电路分成几个可测量的功能模块,静态时可用万用表检查各功能模块的输入和输出,并与正常值做对照。若某个模块的输出有异常,断开负载测量仍不正常,则可认为该模块有问题。做动态检查时可用示波器,依信号流向,逐级观察各模块的输出是否在正常的范围内,找出异常输出后,也要断开负载再测量一次,以判断故障是来自负载的影响,还是模块本身的问题。

③ 检查模块的内部故障。对确定有故障的模块做内部检查,方法与上述方法类似,即逐级观察各单元电路的输入输出,将故障范围进一步缩小到功能单元电路,可通过更换功能单元块排除故障。如果需要把故障定位到元器件级(如电阻、电容、电感和晶体管等),可通过对单元电路各部分电位和波形的测量分析实现。对有反馈回路的电路,查找故障时可将反馈回路断开,使电路成为开环状态;然后利用上面的方法查找并排除故障,当开环电路调试好后再连接反馈回路。

综上所述,电子系统的设计和制作是一个复杂繁琐的过程,由于受课程学时和设备条件的限制,本教材主要介绍数字系统的计算机辅助分析与设计,即借助 EDA 仿真平台进行数字系统的设计和测试。

1.3 EDA 技术简介

一、EDA 技术的内涵

EDA(Electronics Design Automation)即电子设计自动化。现代电子系统设计依靠手工已经无法满足设计要求,设计工作需要计算机上采用 EDA 技术完成。EDA 技术以计算机硬件和系统软件为基本工作平台,采用 EDA 通用支撑软件和应用软件包,在计算机上帮助电子设计工程师完成电路的功能设计、逻辑设计、性能分析、时序测试直至 PCB(印刷电路板)的自动设计等。在 EDA 软件的支持下,设计者完成对系统功能的描述,由计算机软件进行处理得到设计结果。利用 EDA 设计工具,设计者可以预知设计结果,减少设计的盲目性,极大地提高设计的效率。

EDA 通用支撑软件和应用软件包涉及电路与系统、数据库、图形学、图论和拓扑逻辑、计算

数学、优化理论等多学科,EDA 软件的技术指标包括自动化程度、功能完善度、运行速度、操作界面、数据开放性和互换性(不同厂商的 EDA 软件可相互兼容)等。

EDA 技术可涵盖电子电路设计的各个领域:即从低频电路到高频电路、从线性电路到非线性电路、从模拟电路到数字电路、从分立电路到集成电路的全部过程。

二、EDA 技术的发展历程

电子设计自动化在不同的时期有着不同的内容。在 20 世纪 70 年代表现为计算机辅助设计(CAD:Computer Aided Design),即将电子设计中涉及的许多计算采用计算机程序实现。在 20 世纪 80 年代表现为计算机辅助工程(CAE:Computer Aided Engineering),主要体现在一些绘图软件的出现,减轻了设计人员的劳动。从 20 世纪 80 年代末开始,设计复杂程度越来越高。EDA 可以直接根据设计要求,以自顶向下的方式设计,并相应地完成系统描述、仿真、集成和验证等环节,直到最后生成所需要的器件。在以上过程中,除系统级设计和行为级描述及对功能的描述以外均可由计算机自动完成。也就是说,设计人员借助开发软件,可以将设计过程中的许多细节问题抛开,而将注意力集中在产品的总体开发上。这样大大减轻了工作人员的工作量,提高了设计效率,减少了以往复杂的工作,缩短了开发周期,实现了真正意义上的电子设计自动化。这个变化是伴随着片上系统的设计出现的,因此有人将 EDA 转向片上系统看作是一次关于系统设计的革命。

1.4 数字系统概述

一、数字系统的基本概念

数字电子技术基础中介绍的编码器、译码器、数据选择器、加法器、计数器、寄存器等电路,都只能实现某种单一的特定的功能,因此称为功能部件级电路。由若干这样的数字电路和逻辑部件构成的,按一定顺序处理和传输数字信号的设备,称为数字系统。电子计算机、数码照相机等就是常见的数字系统。数字系统从结构上可以划分为控制单元和数据处理单元两部分,如图 1.4.1 所示。



图 1.4.1 数字系统框图

控制单元根据外部输入信号及数据处理单元提供的状态信号,决定下一步要完成的操作,并向数据处理单元发出控制信号以控制其完成该操作。通常以是否有控制单元作为区别功能部件和数字系统的标志,凡是包含控制单元且能按顺序进行操作的系统,不论规模大小,一律称为数

字系统。

数据处理单元接收控制单元发来的控制信号,对输入的数据进行算术运算、逻辑运算、移位操作等处理,然后输出数据,并将处理过程中产生的状态信号反馈到控制单元。数据处理单元也称为数据通路。

二、数字系统设计的一般方法

数字系统设计时,首要任务是明确系统设计的目标,根据设计任务选择和论证方案。总体方案确定后,就可以进行具体设计。一般数字系统有自顶向下、自底向上以及两者相结合的设计方法。

自顶向下的方法是按“系统—子系统—功能模块—单元电路—元器件—印制版图”的流程,即按照由大到小、由粗到细的思路进行设计的。设计时从系统级开始,根据任务和系统指标要求,将系统划分为若干规模适当、功能单一且相对独立的子系统。之后,独立设计每一个系统,确定具体的元器件,并绘制电路原理图和印制电路板图。

自顶向下的设计方法一般要遵循下列几个步骤。

1. 明确所要设计系统的功能,进行逻辑抽象

设计题目通常是比较简单的文字叙述,没有细节说明,设计者必须对题目消化、理解,逐步明确并抽象出系统要完成的逻辑功能。

2. 确定实现系统功能的总体方案,画出系统方框图

将系统要实现的复杂功能进行分解,分成若干子功能模块,并确定各功能模块的操作顺序和相互联系,画出系统的方框图。

3. 设计数据处理单元

明确数据处理单元的基本运算和操作,它们可以是算术和逻辑运算、数据的存储、变换和传送等。选用通用集成电路芯片实现其功能。

4. 设计控制单元

根据数据处理单元进行的操作及操作顺序,确定控制单元的逻辑功能。在绝大多数数字系统中,控制单元是同步时序电路。

自底向上的设计步骤与自顶向下的设计步骤正好相反,这是传统电子系统设计常采用的方法。在现代数字系统设计中一般采用自顶向下的方法,它可同时兼顾设计周期、系统性能和成本。自底向上的设计方法在系统的组装、调试以及以IP核为基础的超大规模片上系统设计中仍得以采用。因此,复杂的电子系统设计实际上是自顶向下和自底向上两种方法互相交织、反复多次的过程。

系统设计无论采用何种方法,都应遵循相应的原则。

(1) 确保每一级设计的正确性和合理性,技术指标应留有余地。

(2) 各子系统之间、模块之间,其功能上应尽量相对独立。

(3) 各层设计中遇到的问题应及时解决,不可以将问题传给下层。如果本层解决不了,必须将问题反馈到上层,在上层中解决。

(4) 软件、硬件协同设计,充分利用微控制器和可编程逻辑器件的可编程功能,在软件、硬件之间寻找平衡。

1.5 数字系统设计实例

在进行智力竞赛抢答比赛时,各参赛者考虑好后都想抢先回答,如果没有合适的设备,主持人将难以分辨出抢答者的先后顺序,为了使比赛能顺利进行,需要有一个能判断抢答者先后顺序的设备,称为智力竞赛抢答器。此处要求设计一个可容纳4个队参加比赛的四路抢答器。

一、明确系统的功能

通过分析题目,智力竞赛抢答器系统要完成的逻辑功能如下。

第一步:主持人预置每组100分。

第二步:主持人按准备按钮,宣布准备抢答并述题。

第三步:判断有无违章提前抢答者。如果有违章提前抢答者(选手抢答台红灯亮并发出声音),则宣布本题失败并做违章处理(如减分或口头警告),然后返回第二步。如果没有违章提前抢答者,则进行第四步。

第四步:主持人按开始抢答按钮,宣布开始抢答。

第五步:选手按自己的选手抢答台上所设的抢答按钮抢答。抢到答题权的选手抢答台上所设的绿灯发亮并发出声音,经主持人允许后答题。

第六步:主持人宣布选手得分,并按加10分按钮给选手加10分或者按减10分按钮给选手减10分。

第七步:转入第二步进行下一题,或宣布抢答活动结束。

二、确定系统的组成

系统应包括主持人控制台、选手抢答台、选手计分器三部分。主持人通过主持人控制台上所设按钮向选手抢答台和选手计分器发送抢答允许、加分、减分等信号;抢答选手通过选手抢答台上所设按钮发送抢答信号并封锁竞争对手的抢答器,同时向自己的计分器发出计分允许信号。

三、主持人控制台功能规划和电路设计

1. 功能规划

(1) 主持人控制台状态和所需器件

主持人控制台只有准备状态 S_0 和抢答开始状态 S_1 ,故只需一个状态触发器 FF_0 ;此外需要预置100分按钮(L)、准备抢答按钮(R)、开始抢答按钮(S)、加10分按钮(U)、减10分按钮(D)。

(2) 按钮功能规定和信号形式

L :同时将四个选手的计分器置成100分,选用带低电平有效预置端的计数器构成计分器,故信号形式为负脉冲。

R :将本台状态触发器 FF_0 复位到 S_0 状态,同时将四个选手抢答台置于初始状态,信号形式为负脉冲。

S :将本台状态触发器 FF_0 置位到 S_1 状态,再由 FF_0 向四个选手抢答台输出抢答允许信号,信号形式为正脉冲。

U :给竞答选手加 10 分,信号形式为正脉冲。

D :给竞答选手减 10 分,信号形式为正脉冲。

2. 电路设计

本题目使用 Multisim 软件中提供的元器件来实现。 FF_0 选用带低电平预置和清零的 D 触发器,将 D 端和 CP 端接地,作为 RS 触发器使用。如果没有按钮,也可以改用单刀双掷开关构成,每次操作必须连接两次才可模仿一个按钮。由此得出电路设计如图 1.5.1 所示。由于 FF_0 在上电仿真时会自动进入置位状态 S_1 ,故在电路中加入 0.1s 的延迟继电器使 FF_0 上电复位。

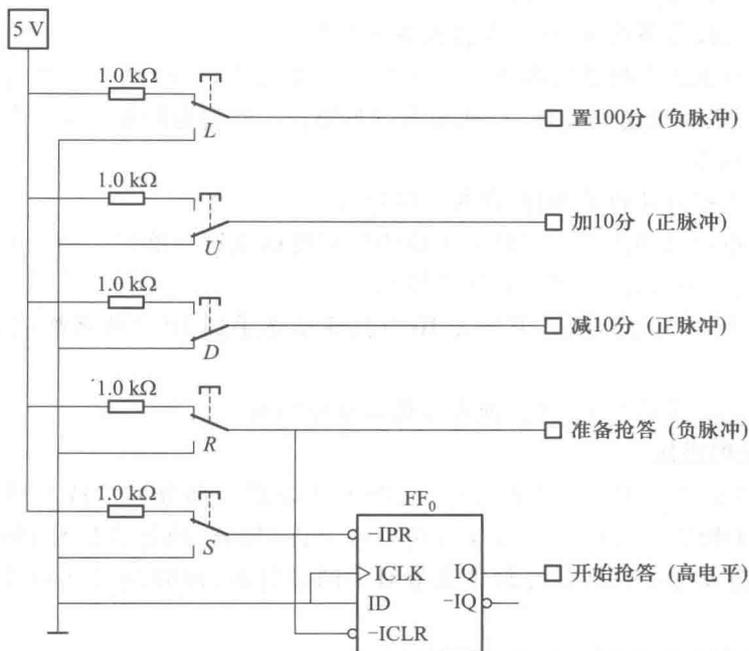


图 1.5.1 主持人控制台电路

四、选手抢答器功能规划和电路设计

1. 选手抢答器的输入和输出信号及其含意

(1) 选手抢答器输入信号应包括:① 主持人控制台送来的准备抢答信号,将选手抢答器复位到初始状态,设为 X_0 ;② 主持人控制台送来的开始抢答信号,允许竞答选手按抢答按钮,设为 X_1 ,1 有效;③ 其他选手抢答器送来的封锁信号,使本选手不能抢答,设其组合

后为 X_2 , 0 封锁; ④ 竞答选手按抢答按钮形成的抢答信号, 在电路中作为时钟信号 CP , 设为 X_3 , 为负脉冲。

(2) 选手抢答器输出信号应包括: ① 违章抢答信号, 用于点亮红灯并使蜂鸣器发声, 设为 Y_1 , 1 有效; ② 抢答有效信号, 用于点亮绿灯, 设为 Y_2 , 1 有效; ③ 计分允许信号, 使主持人能对本选手加分或减分, 设为 Y_3 , 1 有效; ④ 输出封锁信号, 使其他竞答选手按抢答按钮无效, 设为 Y_4 , 1 有效。

2. 选手抢答器的状态规划和设计过程

选手抢答器应至少具有三个状态, 一为初始状态 S_0 , 二为违章抢答后状态 S_1 , 三为有效抢答后状态 S_2 。根据题意可画出原始状态图如图 1.5.2 所示。

在准备抢答信号到来后选手抢答器被复位到初始状态 S_0 , 此时若无抢答允许信号, 则选手按抢答按钮为违章抢答, 状态转到 S_1 并给出红灯信号、计分允许信号和输出封锁信号。此后状态停在 S_1 等待被复位。

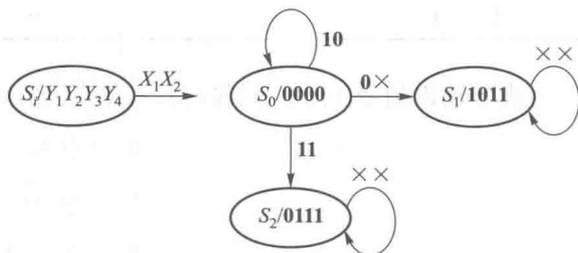


图 1.5.2 选手抢答器的原始状态图

在状态 S_0 下, 若有抢答允许信号并且无输入封锁信号, 则选手按抢答按钮有效, 状态转到 S_2 , 同时给出绿灯信号、计分允许信号和输出封锁信号。此后状态停在 S_2 等待被复位。

在状态 S_0 下, 若有抢答允许信号并且有输入封锁信号, 则选手按抢答按钮无效, 状态停在 S_0 不变, 输出也不变。

根据原始状态图, 可以得到原始状态表, 如表 1.5.1 所示。

表 1.5.1 原始状态表

S^n	X_1X_2			
	00	01	10	11
S_0	$S_1/1011$	$S_1/1011$	$S_0/0000$	$S_2/0111$
S_1	$S_1/1011$	$S_1/1011$	$S_1/1011$	$S_1/1011$
S_2	$S_2/0111$	$S_2/0111$	$S_2/0111$	$S_2/0111$

从原始状态表中可以看出, 计分允许信号 Y_3 和输出封锁信号 Y_4 始终相同, 可以合并 Y_3 和 Y_4 ; 三个状态的输出各异, 已不可化简, 需用两个触发器, 选用带有低电平直接置零端的负边沿双 JK 触发器 74X112。然后对三个状态进行编码分配, S_0 、 S_1 、 S_2 的编码分别选为 00 、 01 、 10 , 设状态 11 为 S_3 , 并规定在任何输入下其输出全为 0 并且次状态为 S_0 。由此得驱动函数和输出函数真值表如表 1.5.2 所示。

表 1.5.2 驱动函数和输出函数真值表

Q_1^n Q_0^n	X_1 X_2	Q_1^{n+1} Q_0^{n+1}	J_1 K_1 J_0 K_0	Y_1 Y_2 Y_3
0 0	0 ×	0 1	0 × 1 ×	0 0 0
	1 0	0 0	0 × 0 ×	
	1 1	1 0	1 × 0 ×	
0 1	× ×	0 1	0 × × 0	1 0 1
1 0	× ×	1 0	× 0 0 ×	0 1 1
1 1	× ×	0 0	× 1 × 1	0 0 0

由于输出信号 Y_1 、 Y_2 、 Y_3 仅与状态 Q_1 、 Q_0 有关,可直接写出其表达式如下

$$Y_1 = \bar{Q}_1^n Q_0^n$$

$$Y_2 = Q_1^n \bar{Q}_0^n$$

$$Y_3 = Y_1 + Y_2 = Q_1^n \oplus Q_0^n$$

根据上表可得各驱动函数的卡诺图如图 1.5.3 所示。

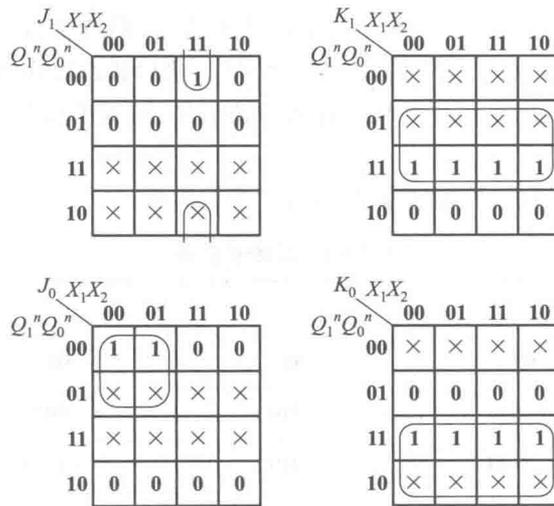


图 1.5.3 驱动函数的卡诺图

从卡诺图得逻辑表达式

$$J_1 = \bar{Q}_0^n \cdot X_1 \cdot X_2$$

$$K_1 = Q_0^n$$

$$J_0 = \bar{X}_1 \cdot \bar{Q}_1^n$$

$$K_0 = Q_1^n$$

输入封锁信号 X_2 为 0 时起封锁作用,其他选手抢答器中任一输出封锁信号为 1 时, X_2 都应为 0,故可用或非门实现。

3. 电路设计

根据上述分析和设计可得电路原理图,如图 1.5.4 所示。

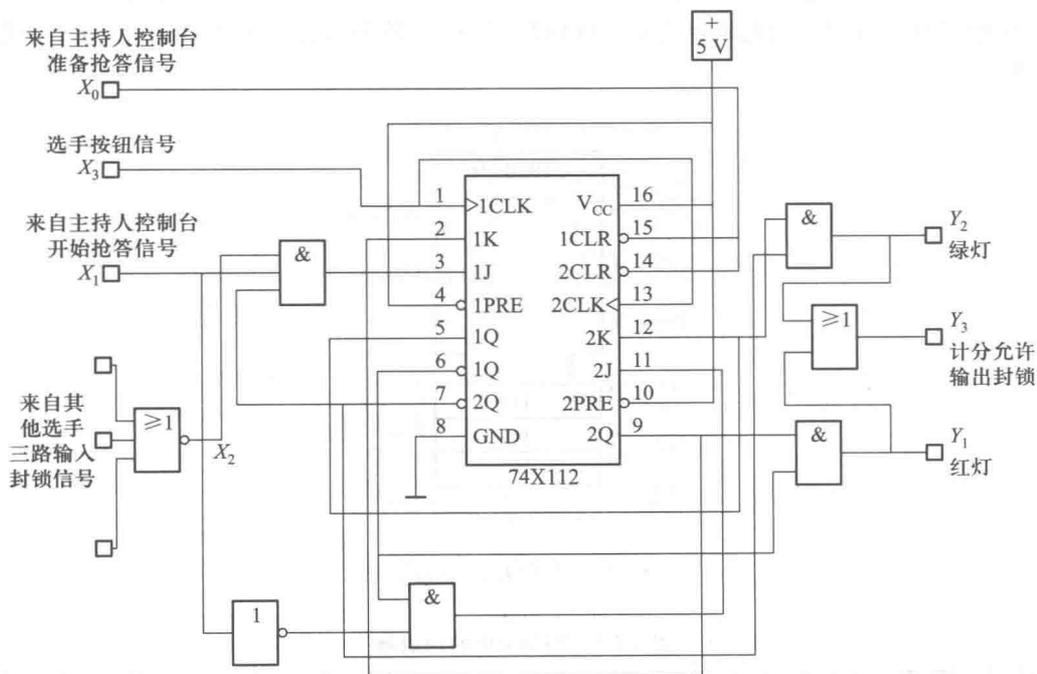


图 1.5.4 选手抢答器电路

由于同样的电路需要 4 套,全部画出来使得总图太大,故而可用 Multisim 所提供的生成子电路的方法将其生成一个小的模块,如图 1.5.5 所示,以使电路总图简洁并便于仿真。注意生成子电路后一定要记住各引出脚的含意,以便和总图中的其他电路正确连接。



图 1.5.5 选手抢答器电路接口信号示意图

五、计数器功能和电路设计

1. 计数器应具备基本功能

- (1) 计数范围为 000~990,以 10 为最小计数单位;
- (2) 采用 3 位七段显示数码管显示计分值;

- (3) 可用预置脉冲将计数值预置成 100 分；
- (4) 具有计分允许信号输入端,该信号有效时方可对本计数器进行加分或减分操作；
- (5) 具有加 10 分脉冲信号和减 10 分脉冲信号输入端。

2. 电路设计

根据计分器的功能要求,应选用具有清零和置数功能的十进制可逆计数器,查阅资料《中国集成电路大全》之 TTL 电路分册,有 74X168 和 74X192 两种芯片可以选用。但在 Multisim 的数字器件库中只有 74X192,因此确定选用 74X192,74LS192 的逻辑符号如图 1.5.6 所示,功能表如表 1.5.3 所示。

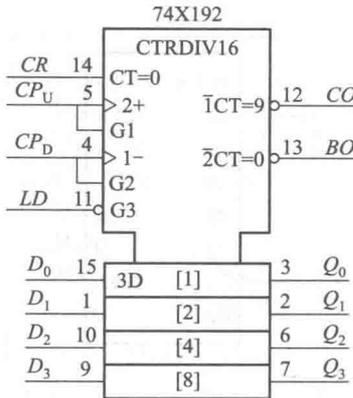


图 1.5.6 74LS192 的逻辑符号

表 1.5.3 74LS192 的功能表

清零	预置	“加” 计数时钟	“减” 计数时钟	预置数据输入	输出	工作模式
CR	LD	CP _U	CP _D	D ₃ D ₂ D ₁	Q ₃ Q ₂ Q ₁ Q ₀	
1	x	x	x	x x x x	0 0 0 0	异步清零
0	0	x	x	D C B	D C B	异步置数
0	1	1	1	x x x x	保持	数据保持
0	1	↑	1	x x x x	计数	加法计数
0	1	1	↑	x x x x	计数	减法计数

根据所选器件可构成计分器电路如图 1.5.7 所示,本设计需要 4 套相同电路,同样生成子电路。

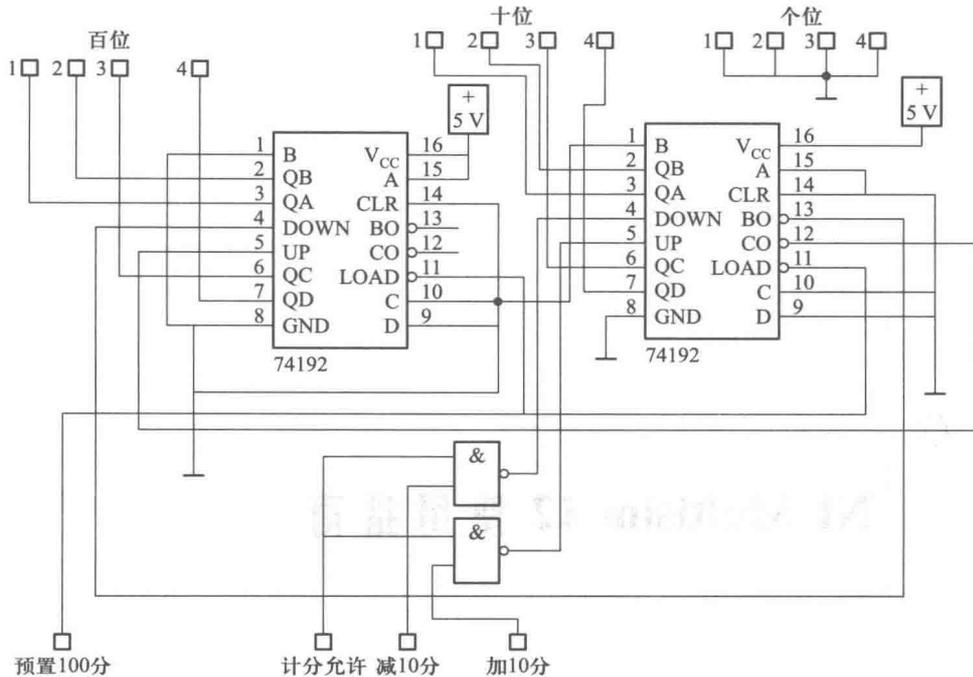


图 1.5.7 计数器电路

六、整体连接和调试

在 Multisim 环境下将主持人控制台、四路选手抢答器、四路计分器互相正确连接,加入必要的外部元件如开关、红绿灯、蜂鸣器、数码管等,基本完成了抢答器系统的设计,然后便可接通电源进行仿真调试。仿真时应按照设计要求逐条检验,看所设计的电路是否完全具备预想的所有功能。通常一次取得完全成功的可能性不大,这时候需要将原设计作某些调整或改变,直到电路完全满足设计要求。

七、撰写设计报告

设计报告是对整个设计题目的总结,应根据综合实验的目的与要求,设计框图及电路系统概述,各单元电路的设计方案及原理说明、参数计算,调试过程及结果分析,设计、安装及调试中的体会,参考文献等进行分项描述。

设计报告一人一份独立完成,要求有理论分析、要实事求是、字迹要清楚、文理要通顺、图表要整洁、结论要明确。

第二章

NI Multisim 12 使用指南

2.1 NI Multisim 12 简介

EDA 是电子设计自动化(Electronic Design Automation)的简称,是现代电子工程领域的一门新技术,它提供了基于计算机和信息技术的电路系统设计方法。

传统的电子电路与系统设计方法周期长、耗材多、效率低,难以满足电子技术飞速发展的要求。引入 EDA 技术改变了以定量估算和电路实验为基础的电路设计方法,极大地提高了电子电路与系统的设计质量和效率。熟练掌握和运用 EDA 技术是电类专业的基本要求,也是当今电子电路分析与设计人员必须具备的基本技能之一。

EDA 技术和设计手段包括使用硬件描述语言表达设计意图、以大规模集成器件做硬件载体、以计算机为设计开发工具以及以 EDA 软件为开发环境的现代电子技术方法的综合。

随着 EDA 技术的发展,其涉及的内容也日趋庞大,不仅涉及用计算机技术的某几种软、硬件开发系统的方法来完成硬件设计和实现,而且还与现代自动化设计中行为与结构综合的概念、自顶向下的设计理念等内容相结合。

从 20 世纪 80 年代开始,随着计算机技术的飞速发展,电子电路的分析与设计方法发生了重大变革,Pspice、EWB 等一大批各具特色的优秀电子设计自动化(EDA)软件的出现,改变了以定

量估算和电路实验为基础的电路设计方法。熟练掌握一些电路仿真软件已成为当今电子电路分析和设计人员所必须具备的基本技能之一。

NI Multisim 12 是美国国家仪器有限公司推出的 NI Circuit Design Suite 12 中的一个重要组成部分,其前身为 EWB(Electronics Work-bench)。NI Multisim 是一种交互式电路模拟软件,是一种 EDA 工具,它为用户提供了丰富的元件库和功能齐全的各类虚拟仪器,主要用于对各种电路进行全面的仿真分析和设计。NI Multisim 提供了集成化的设计环节,能完成原理图的设计输入、电路仿真分析、电路功能测试等工作。当需要改变电路参数和电路结构仿真时,可以清楚地观察到各种变化电路对性能的影响。用 NI Multisim 进行电路的仿真,实验成本低、速度快、效率高。

NI Multisim 12 包含了数量众多的元器件库和标准化的仿真仪器库,用户还可以自己添加新元件,操作简单,分析和仿真功能十分强大。熟练使用该软件可以大大缩短产品研发的时间。

NI Multisim 12 的特点如下。

1. 直观的图形界面

NI Multisim 12 的整个界面就像是一个电子实验工作平台,绘制电路所需的元器件和仿真所需的仪器仪表均可直接拖放到工作区中,轻点鼠标即可完成导线的连接,软件仪器的控制面板和操作方式与实物相似,测量数据、波形和特性曲线如同在真实仪器上看到的一样。

2. 丰富的元器件库

NI Multisim 12 具有丰富的元器件库,包括基本元器件、半导体元器件、TTL 以及 CMOS 数字 IC、DAC、ADC、MCU 和其他各种部件,且用户可通过元件编辑器自行创建或扩充已有的元器件库,而且所需的元器件参数可以从生产厂商的产品手册中查到,因此很方便在工程设计中使用。

3. 丰富的测试仪器仪表

NI Multisim 12 除了具备一般实验用的通用仪器,如数字万用表、函数信号发生器、双踪示波器、直流电源,还有实验室少有或没有的仪器,如波特图仪、字信号发生器、逻辑分析仪、逻辑转换器、失真分析仪、频谱分析仪和网络分析仪等,且增加了安捷伦的信号源、万用表、示波器及泰克的示波器等,所有仪器均可多台同时调用。

4. 完备的分析手段

NI Multisim 12 具有详细的电路分析功能,可以完成电路的瞬态分析和稳态分析、时域分析和频域分析、器件的线性 and 非线性分析、电路的噪声分析和失真分析、离散傅里叶分析、电路零点分析、交直流灵敏度分析等电路分析方法,可以在线显示图形并具有很大的灵活性,以帮助设计人员分析电路的性能。

5. 强大的仿真能力

NI Multisim 12 可以设计、测试和演示各种电子电路,包括电工学、模拟电路、数字电路、射频电路及微控制器和接口电路等。可以对电路设置各种故障,如开路、短路和不同程度

的漏电等,从而观察不同故障情况下的电路工作状况。在进行仿真时,还可以存储测试点的所有数据,列出电路的元器件清单,以及存储测试仪器的的工作状态、显示波形和具体数据等。

6. 完美的兼容能力

NI Multisim 12 可方便地将模拟结果以原有文档格式导入 Lab VIEW 或者 Signal Express 中。工程人员可更有效地分享及比较仿真数据和模拟数据,而无需转换文件格式,在分享数据时减少了失误,提高了效率。

7. 丰富的在线帮助

NI Multisim 12 有丰富的 Help 功能,不仅包括软件本身的操作指南,更重要的是包含元器件的功能解说,这种功能解说有利于使用 EWB 进行 CAI 教学。另外,NI Multisim 12 还提供了与国内外流行的印刷电路板设计自动化软件 Protel 及电路仿真软件 PSpice 之间的文件接口,也能通过 Windows 的剪贴板把电路图送往文字处理系统中进行编辑排版。支持 VHDL 和 Verilog HDL 语言的电路仿真与设计。

8. 高效的电路设计

利用 NI Multisim 12 可以实现计算机仿真设计与虚拟实验,与传统的电子电路设计与实验方法相比,具有如下特点:设计与实验可以同步进行,可以边设计边实验,修改调试方便;设计和实验用元器件及测试仪器仪表齐全,可以完成各种类型的电路设计与实验;可方便地对电路参数进行测试和分析;可直接打印输出实验数据、测试参数、曲线和电路原理图;实验中不消耗实际的元器件,实验所需元器件的种类和数量不受限制,实验成本低,实验速度快,效率高;设计和实验成功的电路可以直接在产品中使用。

NI Multisim 12 易学易用,便于电子信息工程、通信工程、自动化、电气控制类专业学生自学、便于开展综合性的设计和实验,有利于培养学生综合分析能力、开发和创新能力。针对不同的用户需要,NI Multisim 12 发行了多个版本,分为增强专业版(Power Professional)、专业版(Professional)、个人版(Personal)、教育版(Education)、学生版(Student)和演示版(Demo)等。我们这里仅对 NI Multisim 12 教育版进行介绍。

2.2 NI Multisim 12 的基本操作界面

一、主窗口

单击开始—程序—National Instruments—Circuit Design Suite 12.0—Multisim 12.0,启动 NI Multisim 12,弹出如图 2.2.1 所示的界面,即 NI Multisim 12 的基本操作界面,该界面主要由电路工作区、工具栏、仪器仪表栏、元器件栏、仿真开关等组成。这个界面相当于一个虚拟的电子实验平台。

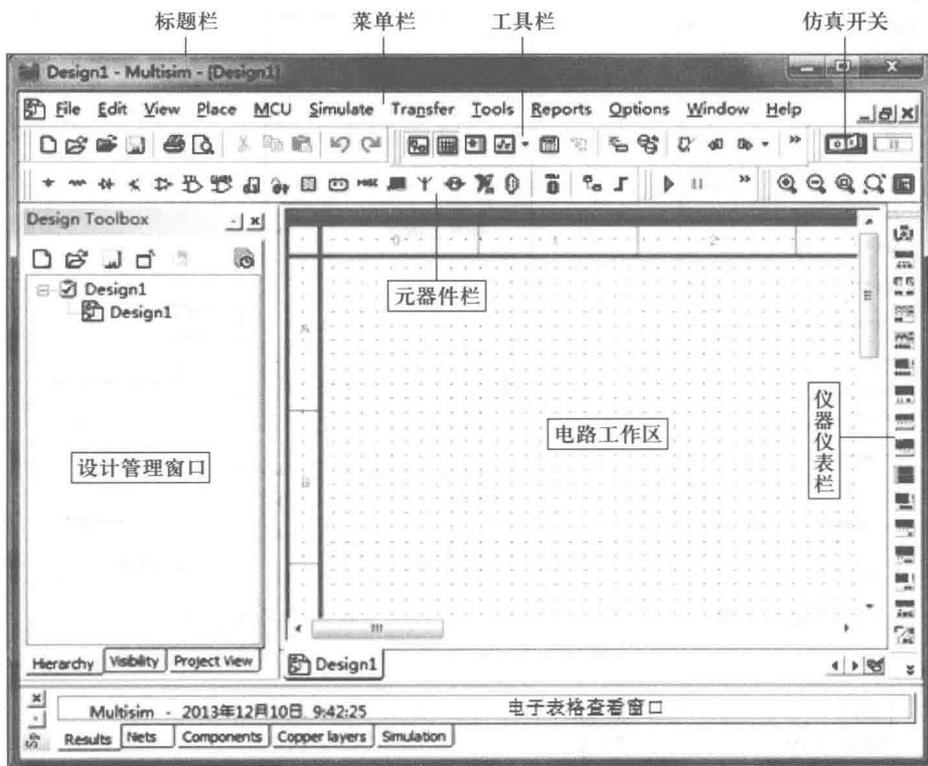


图 2.2.1 NI Multisim 12 的基本操作界面

二、菜单栏

NI Multisim 12 有 12 个菜单项,如图 2.2.2 所示。菜单中提供了软件所有的功能命令。



图 2.2.2 NI Multisim 12 菜单栏

(1) File 文件菜单,如图 2.2.3 所示。File 菜单提供了 17 个文件操作命令,如打开、保存、打印等,主要用于管理所创建的电路文件。

(2) Edit 编辑菜单,如图 2.2.4 所示。Edit 菜单提供对电路和元件进行剪切、粘贴、旋转等 23 个操作命令,主要用于在电路绘制过程中对电路和元器件进行各种技术性处理。

(3) View 视图菜单,如图 2.2.5 所示。View 菜单提供了 21 个用于控制仿真界面上显示的内容、缩放电路原理图和查找元件等操作命令。

(4) Place 放置菜单,如图 2.2.6 所示。Place 菜单提供了在电路工作窗口内放置元件、节点、导线、各种连接接口及文本等命令。

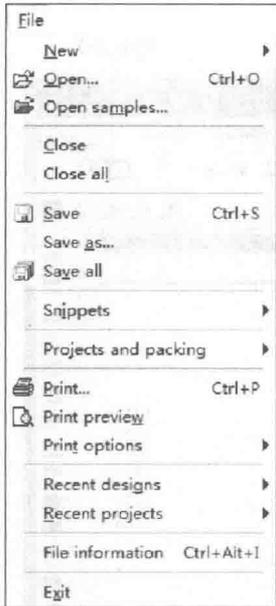


图 2.2.3 文件菜单

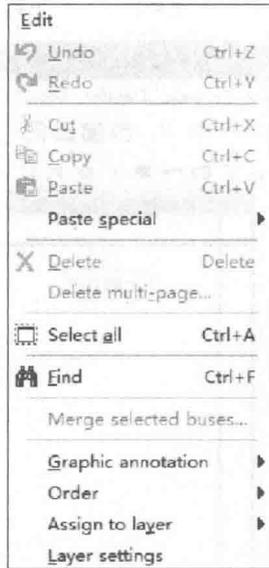


图 2.2.4 编辑菜单



图 2.2.5 视图菜单

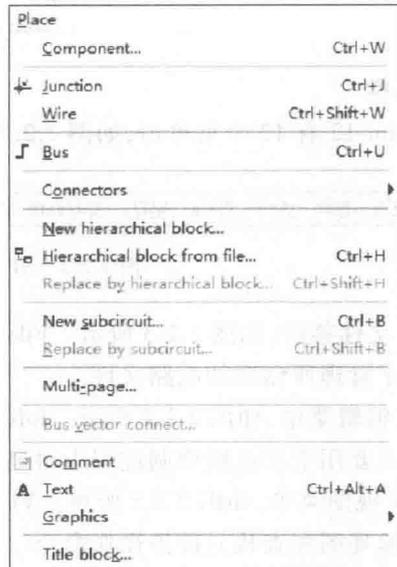


图 2.2.6 放置菜单

(5) MCU 微控制器菜单,如图 2.2.7 所示。MCU 菜单提供了带有微控制器的嵌入式电路仿真功能。所支持的微控制器芯片类型有两类:80C51 和 PIC。

(6) Simulate 仿真菜单,如图 2.2.8 所示。Simulate 菜单提供了常用的仿真设置与操作命令。

(7) Transfer 文件输出菜单,如图 2.2.9 所示。Transfer 菜单提供仿真电路的各种数据与 Ultiboard 12 数据相互传送的功能。

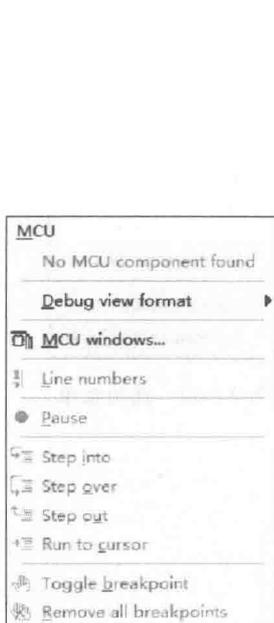


图 2.2.7 微控制器菜单



图 2.2.8 仿真菜单



图 2.2.9 文件输出菜单

(8) Tools 工具菜单,如图 2.2.10 所示。Tools 菜单提供了常用电路创建向导和电路管理命令,主要用于编辑和管理元器件和元件库。

(9) Reports 报告菜单,如图 2.2.11 所示。Reports 菜单用于产生指定元件存储在数据库中的所有信息和当前电路窗口中所有元件的详细参数报告。

(10) Options 选项菜单,如图 2.2.12 所示。Options 菜单提供了用户需要设置电路功能、存储模式及工作界面功能。

(11) Window 窗口菜单,如图 2.2.13 所示。Window 菜单提供了对一个电路的各个多页面电路以及不同的各个仿真电路同时浏览的功能。

(12) Help 帮助菜单,如图 2.2.14 所示。Help 菜单为用户提供在线技术帮助和使用指导,包含帮助主目录、帮助主题索引及版本说明等选项。



图 2.2.10 工具菜单

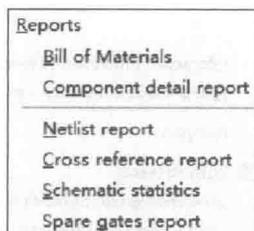


图 2.2.11 报告菜单

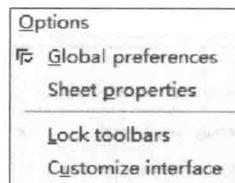


图 2.2.12 选项菜单

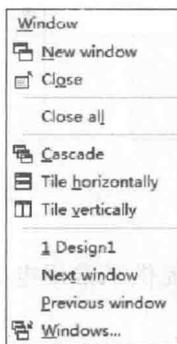


图 2.2.13 窗口菜单

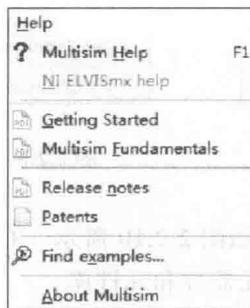


图 2.2.14 帮助菜单

三、工具栏

Multisim 的工具栏主要包括 Standard Toolbar(标准工具栏)、Main Toolbar(系统工具栏)、View Toolbar(视图工具栏)、Component Toolbar(元件库)、Virtual Toolbar(虚拟元件库)、Graphic Annotation Toolbar(图形注释工具栏)、Status Toolbar(状态栏)和 Instrument Toolbar(仪器仪表工具栏)等,如图 2.2.1 所示。若需打开相应的工具栏,可通过单击 View→Toolbars 菜单项,在弹出的级联子菜单中即可找到相应项。

2.3 元器件库

Multisim 将所有的元器件模型分门别类地放到 18 个分类库中,每个元器件库放置同一种类型的元器件,如图 2.3.1 所示。



图 2.3.1 元器件库

一、电源/信号源库

电源/信号源库共有 6 个系列,包含接地端、直流电源、交流电源、时钟电源、受控电源等 74 种电源与信号源,如图 2.3.2 所示。

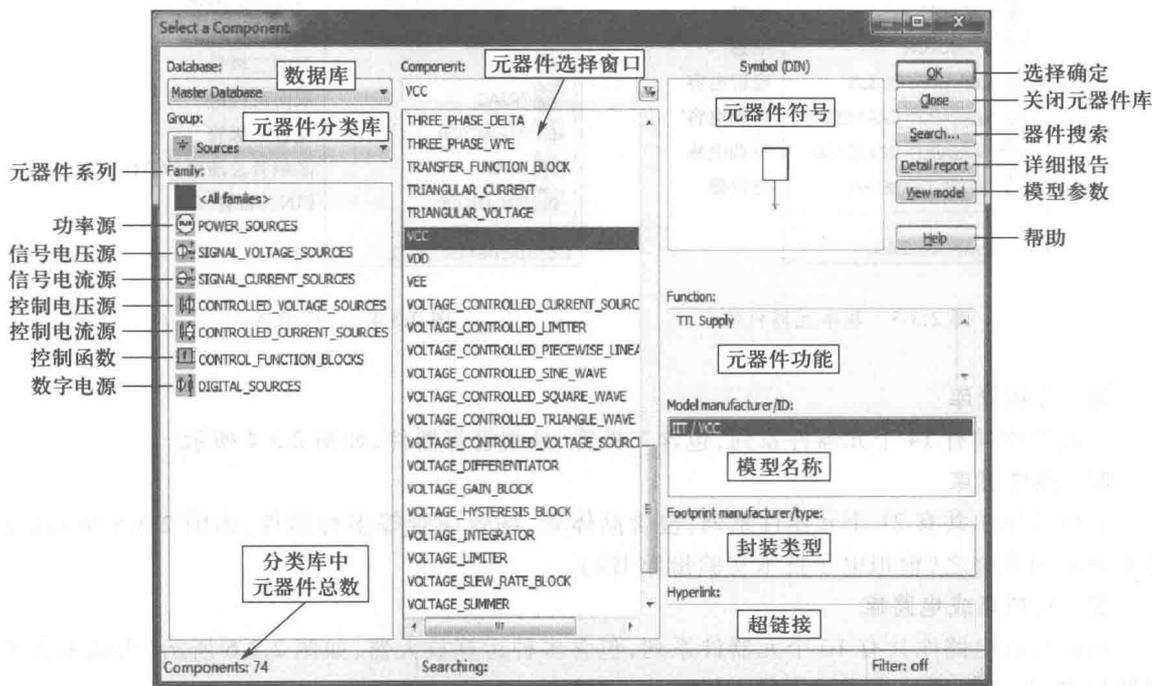


图 2.3.2 电源/信号源库

二、基本元器件库

基本元器件库共有 16 个元器件系列,包含电阻、电容、电感等基本元件。基本元器件库中的虚拟元器件的参数可以任意设置,非虚拟元器件的参数是固定的,但可以根据需要选择,如图 2.3.3 所示。

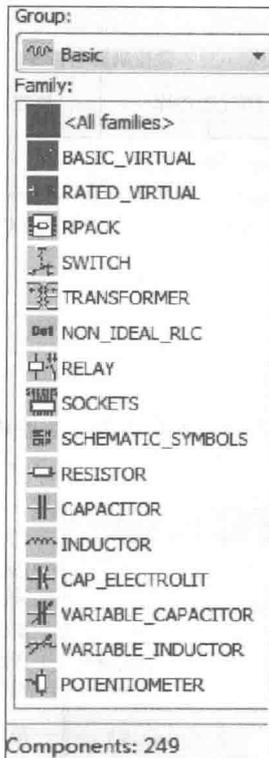


图 2.3.3 基本元器件库

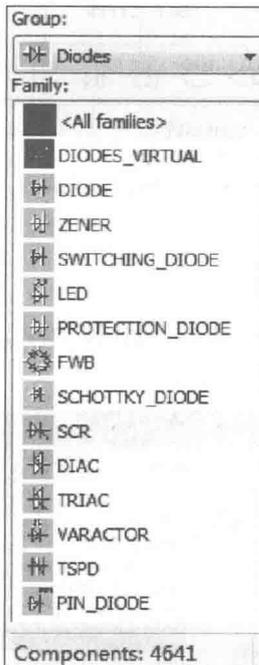


图 2.3.4 二极管库

三、二极管库

二极管库共有 14 个元器件系列,包含二极管、晶闸管等器件,如图 2.3.4 所示。

四、晶体管库

晶体管库中共有 21 个元器件系列,包含晶体管、场效应管等多种器件,如图 2.3.5 所示(参阅本套系列教材之《模拟电子技术实验指导书》)。

五、模拟集成电路库

模拟集成电路库共有 10 个元器件系列,包含多种运算放大器,如图 2.3.6 所示(参阅本套系列教材之《模拟电子技术实验指导书》)。

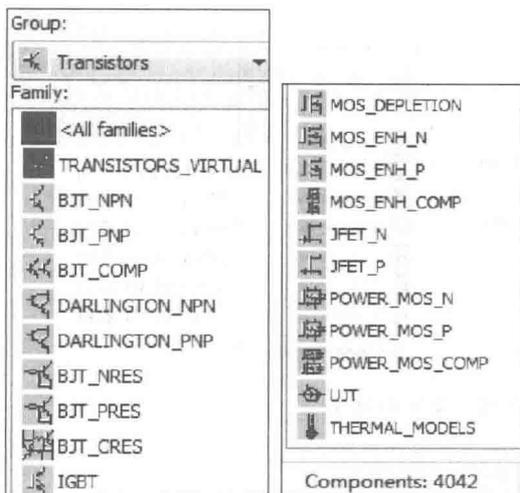


图 2.3.5 晶体管库

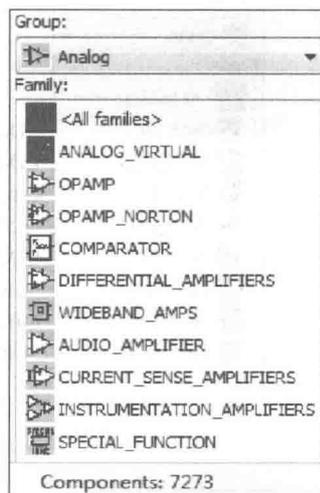


图 2.3.6 模拟集成电路库

六、TTL 数字集成电路库

TTL 数字集成电路库共有 9 个元器件系列,包含 74 系列、74S 系列、74LS 系列、74F 系列等 74 系列数字集成电路,如图 2.3.7 所示。



图 2.3.7 TTL 数字集成电路库

七、CMOS 数字集成电路库

CMOS 数字集成电路库共有 11 个系列,包含有 4000 系列和 74HC 系列多种 CMOS 数字集成电路,如图 2.3.8 所示。

八、数字元器件库

数字元器件库共有 14 个系列,包含有 4000 系列和 74HC 系列多种 CMOS 数字集成电路,如图 2.3.9 所示。



图 2.3.8 CMOS 数字集成电路库

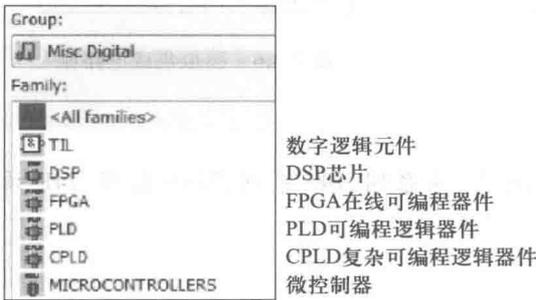


图 2.3.9 数字元件库



图 2.3.10 混合集成电路库

九、混合集成电路库

混合集成电路库共有 6 个元器件系列,包含 555 定时器、多谐振荡器等多种数模混合集成电路,如图 2.3.10 所示。

十、指示器件库

指示器件库共有 8 个元器件系列,包含 8 种可用来显示电路仿真结果的显示器件,如图 2.3.11所示。

十一、功率电源库

功率电源库共有 12 个系列,包括三段稳压器、开关电源等多种功率电源,如图 2.3.12 所示。

十二、其他器件库

其他器件库共有 14 个系列,包含光电耦合器、晶振、滤波器等多种器件,如图 2.3.13 所示。



图 2.3.11 指示器件库



图 2.3.12 功率电源库



图 2.3.13 其他器件库

十三、外围设备库

外围设备库共有 4 个系列,包含键盘、液晶屏等器件,如图 2.3.14 所示。

十四、射频元器件库

射频元器件库共有 8 个系列,包含射频晶体管、射频 FET 等射频元器件,如图 2.3.15 所示。

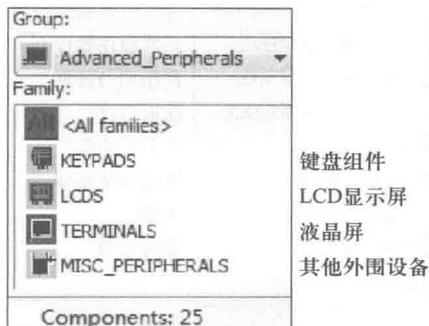


图 2.3.14 外围设备库



图 2.3.15 射频元器件库

十五、机电类器件库

机电类器件库共有 8 个系列,包含传感器、继电器等机电类器件,如图 2.3.16 所示。

十六、NI 元器件库

NI 元器件库共有 9 个系列,包含数据采集卡、信号调理模块等,如图 2.3.17 所示。



图 2.3.16 机电类器件库



图 2.3.17 NI 元器件库

十七、连接器

连接器共有 12 个系列,包含不同厂家的常用接插件,如图 2.3.18 所示。

十八、微控制器

控制器件库共有 4 个系列,包含 805X 系列单片机、存储器等,如图 2.3.19 所示。

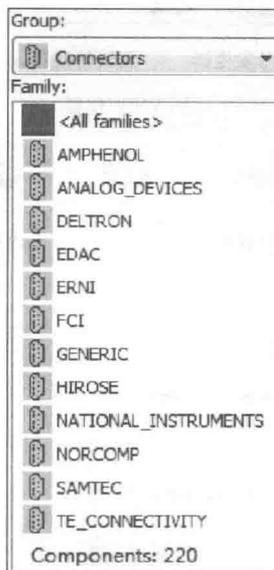


图 2.3.18 连接器



图 2.3.19 微控制器

2.4 仪器仪表库

Multisim 中的仪器仪表是一种具有虚拟面板的计算机仪器,主要由计算机和控制软件组成。操作人员通过图形用户界面用鼠标或键盘来控制仪器运行,以完成对电路的电压、电流、电阻及波形等物理量的测量。虚拟仪器与实际的仪器仪表的操作非常相似,这使仿真实验的操作更加直观、方便。

NI Multisim 12 的仪器仪表库存放有 20 多种虚拟仪器,包括数字万用表、函数信号发生器、示波器(模拟示波器和数字示波器)、波特图仪、字信号发生器、逻辑分析仪、瓦特表、失真度分析仪、频谱分析仪、网络分析仪、测量探针、Lab VIEW 仪器以及电流探针等,如图 2.4.1 所示。每种虚拟仪器都有图标、符号图和仪器面板 3 个外观视图,如图 2.4.2 所示。

图标(Icon):用于在仿真电路图中放置仪器和仪器与测试点的连接,图标的接线端子与电路相连;在仿真状态或者绘制电路图状态下,双击图标即可打开或者关闭仪器面板。

符号(Symbol):用于在工具栏中代表相应的虚拟仪器。

仪器面板(Panel):用户与仪器的交互界面,用于设置参数和显示数据。



图 2.4.1 仪器仪表库

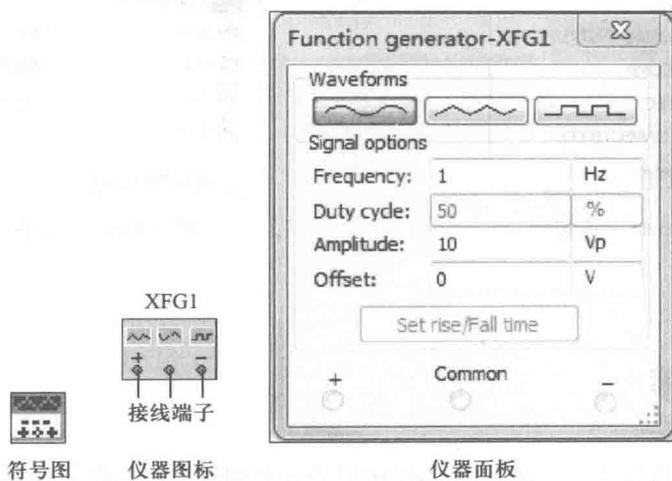


图 2.4.2 虚拟仪器符号、图标和面板

一、虚拟仪器的选用与连接

虚拟仪器仪表的基本操作方法如下。

- (1) 仪器选用:从仪器仪表库中将所用的仪器图标,用鼠标拖放到电路图工作区即可。
- (2) 仪器连接:将仪器图标上的接线端与相应电路的连接点连接。
- (3) 仪器参数设置:双击仪器图标打开仪器面板,用鼠标点击仪器面板上相应的按钮、旋钮和参数设置对话框完成仪器仪表的参数设置。
- (4) 仿真运行:打开仿真电源开关后,可观测数据或观察波形。

在同一电路中可以使用一种多台或多种多台虚拟仪器仪表,而且在仿真过程中,可以根据实际需要实时修改仪器的参数以满足实验要求。

二、虚拟仪器仪表的功能和使用

1. 数字万用表(Multimeter)

数字万用表是一种可以用来测量交直流电压、电流、电阻及电路中两点间分贝损耗的多功能仪表。数字万用表的图标和面板如图 2.4.3 所示,面板上有 1 个数字显示窗口和 7 个按钮,它们分别为电流(A)、电压(V)、电阻(Ω)、衰减(dB)、交流(~)、直流(-)和参数设置(Set...)按钮,单击这些按钮便可进行相应的转换。

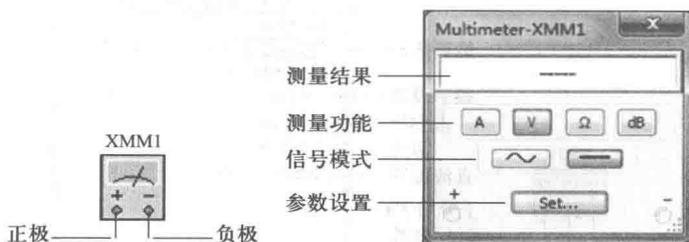


图 2.4.3 数字万用表的图标和面板

仿真平台上的数字万用表具有自动量程转换功能,因此不用指定测量范围。利用参数设置按钮 Settings,打开参数设置对话框,如图 2.4.4 所示,可调整电流表内阻(Ammeter resistance)、电压表内阻(Voltmeter resistance)、欧姆表电流(Ohmmeter current)和电平表的相对分贝值(dB relative value)。其中,电流表(A)量程:0.01 μ A~999 kA;电压表(V)量程:0.01 μ V~999 kV;欧姆表(Ω)量程:0.001 Ω ~999 M Ω ;交流频率范围:0.001 Hz~9 999 MHz。

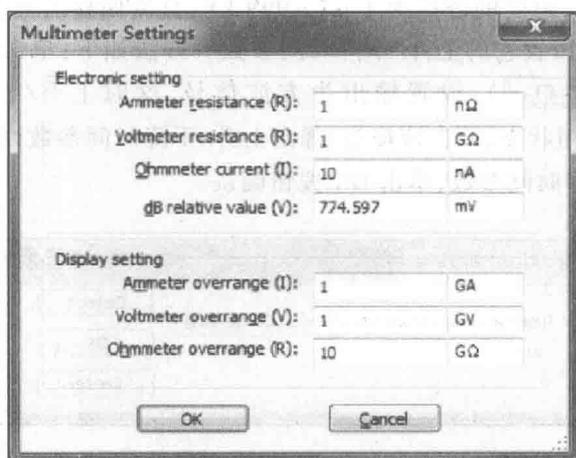


图 2.4.4 数字万用表参数设置对话框

根据所测量信号的特点,在万用表面板上选择信号模式。

交流信号():此时测量的是均方根电压(RMS)或交流信号的电流,直流信号无法进入万用表,只有交流成分会被测量到。

直流信号():此时测量的是直流电压或直流电流信号。

2. 函数信号发生器(Function generator)

函数信号发生器是一种能提供正弦波、三角波或方波信号的电压源,可以方便而又不失真的向电路提供信号。函数信号发生器的图标和面板如图 2.4.5 所示。

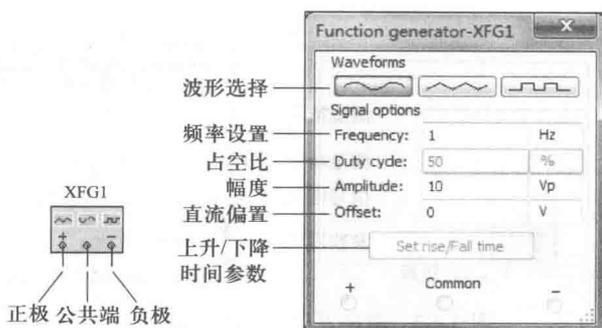


图 2.4.5 函数信号发生器的图标和面板

虚拟信号发生器的图标有三个端子:“+”为正波形端、“COM”为接地端、“-”为负波形端。虚拟仪器发生器的使用方法与实际信号发生器基本相同。

函数信号发生器的输出波形(Waveforms)、工作频率(Frequency)、占空比(Duty cycle)、幅度(Amplitude)和直流偏移(Offset)等参数可以调节。其中,频率的范围为 1 Hz~999 THz;占空比调整值为 1%~99%,幅度的范围为 1 μ V~999 kV;直流偏移的范围为-999 kV~999 kV。

对于方波信号,还可设置它的上升/下降时间参数。方法如下:单击函数信号发生器的波形选择中的方波按钮() ,设置输出为方波信号,这时上升/下降时间参数设置按钮()变为可用状态,单击该按钮,弹出上升/下降时间参数设置对话框,如图 2.4.6 所示,输入设计的上升/下降时间参数,单击 OK 按钮确认。

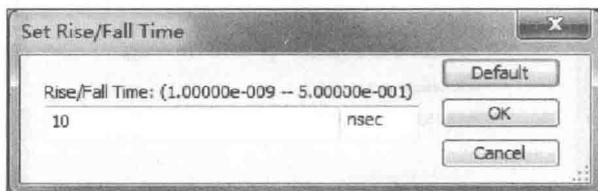


图 2.4.6 上升/下降时间参数设置对话框

3. 瓦特表(Wattmeter)

瓦特表用来测量电路的功率,交流或者直流均可测量。用鼠标双击瓦特表的图标,可以打开它的面板如图 2.4.7 所示。瓦特表的电压端与测量电路并联,电流端与测量电路串联,即可实现测量电路的电压和电流,并显示出乘积,即功率。此外,瓦特表还可以测量功率因子,其取值范围为 0~1。

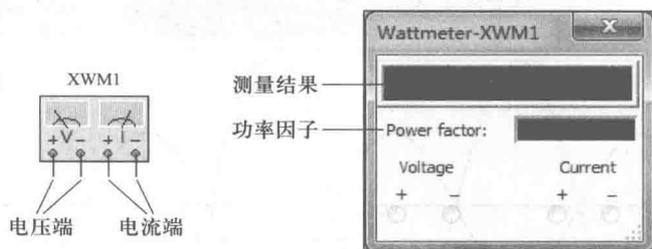


图 2.4.7 瓦特表的图标和面板

4. 两通道示波器(Oscilloscope)

示波器是用来显示信号波形的形状、大小以及频率等参数的仪器。虚拟示波器的面板、参数设置及调整方法与实际示波器类似,但虚拟示波器可以不接地,只要电路中有接地元件即可。

两通道示波器为一种双踪示波器,其图标和面板如图 2.4.8 所示,该仪器的图标上共有 6 个端子,分别为 A 通道的正负端、B 通道的正负端和外触发的正负端。连接时要注意它与现实仪器的不同。

(1) A、B 两个通道的正端分别只需要一根导线与待测点相连接,测量的是该点与地之间的波形。

(2) 若需要测量器件两端的信号波形,只需将 A 或 B 通道的正负端与器件的两端相连即可。

两通道示波器面板各部分的功能如下。

(1) 时基(Time base)控制部分的调整

- 时间基准 Scale:调整示波器的 X 轴的时基值,可选范围为 $1 \text{ fs/Div} \sim 1000 \text{ Ts/Div}$;为了在示波器上得到一个可视性比较好的波形,可设置时基接近信号频率的倒数。例如,输入示波器的信号为 1 kHz ,可以设置 $\text{Scale} = 1 \text{ ms/Div}$ 左右。

- 水平位移 X pos.(Div):调整 X 轴的起始点,范围为 $-5.00 \sim +5.00$;当水平位移调到 0,信号从显示器的左边缘开始。水平位移设置为正数则使起始点右移,设置为负数则使起始点左移。

- 显示方式:可设置示波器的 X、Y 坐标,共有 4 种方式。

① Y/T(幅度/时间)——代表 Y 轴显示 A、B 通道的输入信号,X 轴为时间基线,并按设置时间进行扫描。当要显示时间变化的信号波形时,才采用这种方式。

② Add(A+B)(通道 A+通道 B)——代表 X 轴按设置时间进行扫描,而 Y 轴显示通道 A 与

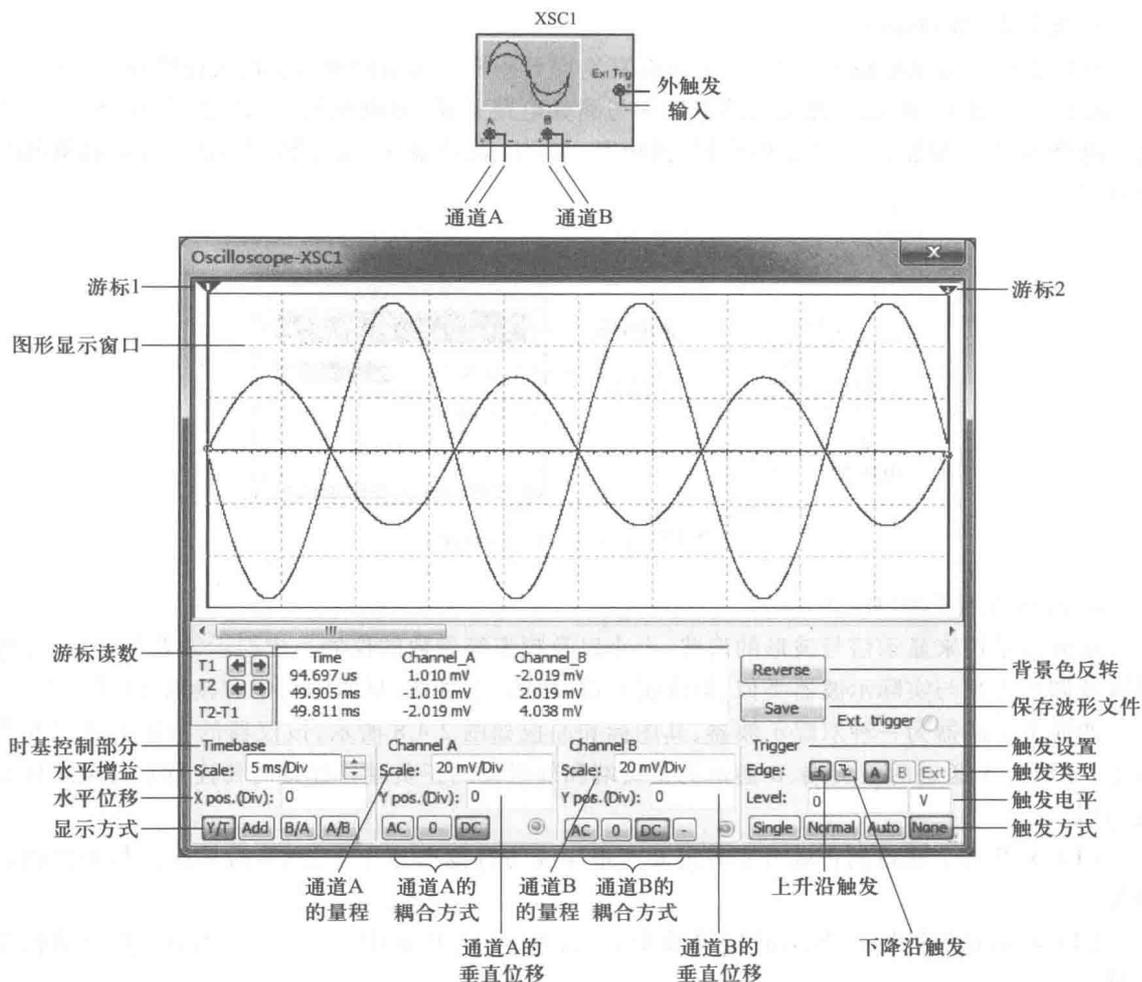


图 2.4.8 模拟双踪示波器的图标和面板

通道 B 的输入信号之和。

③ B/A(通道 B/通道 A)——代表将 A 通道信号作为 X 轴扫描信号,将 B 通道信号施加在 Y 轴上。

④ A/B(通道 A/通道 B)——代表将 B 通道信号作为 X 轴扫描信号,将 A 通道信号施加在 Y 轴上。

(2) 输入通道设置(Channel A/B)

- 量程 Scale:用于选择示波器显示的量程,范围为 1 fV/Div ~ 1 000 TV/Div,可以根据输入信号大小来选择合适的电压量程,使信号波形在示波器显示屏上显示出合适的幅度。

- 垂直位移 Y pos.(Div):是指时间基线在显示屏幕中的上下位置,调整范围为 -99.00 ~

+99.00。当值大于 0 时,时间基线在屏幕中线的上侧,否则在屏幕中线的下侧。通过对通道 A/B 设置不同的垂直位移可以较好地观察和比较两个通道的信号。

- 输入耦合方式:

AC——显示信号的交流分量,这种耦合方式相当于在示波器的探针中串联了一个电容;

DC——显示信号的直流分量和交流分量之和,仿真中一般使用这种方式;

0——信号接地,此时在示波器中出现一条与水平轴重合的参考电平线。

需要注意的是,在通道 B 中的  按钮,可将通道 B 的输入信号进行 180°的相移。

(3) 触发设置(Trigger)

触发设置用于设置示波器的触发类型、触发电平、触发方式等。

- 触发方式(Type):触发信号可以是内部的,即使用通道 A/B 的信号作为触发信号,也可以是外部信号。使用外部触发信号时,可以将触发信号接在示波器面板的“外触发源”接线柱上。触发方式共有以下五种。

Sing——单脉冲触发,当信号达到触发电平时示波器触发一次,当信号显示满一屏时,只有再次单击 Sing 按钮才会显示下一屏信号。

Normal——设置示波器每次到达触发电平时进行刷新。

AUTO——自动触发,示波器自主选择通道信号作为触发信号;一般情况下使用该种方式。

None——普通脉冲触发。

A、B、EXT——选择通道 A、通道 B 或外触发信号作为触发源。

- 触发类型(Edge):可选择上升沿  或下降沿  触发。

- 触发电平(Level):用于设置触发信号的触发门限(单位可选),设置范围为 -999 ~ 999 kV。只有当信号的电平高于触发电平时才会出现在示波器上。

(4) 游标及其读数

示波器有两个游标,用鼠标可以拖动游标,移动过程中游标读数窗口中可观察到游标测量通道信号的读数(包括时间、幅值、时间差和幅值差等)。

另外,单击 Reverse(背景色反转)按钮,可以改变示波器屏幕的背景颜色(白和黑之间转换);单击 Save(保存波形文件)按钮可以以 ASCII 文件形式保存当前示波器显示的波形。

5. 四通道示波器(Four channel oscilloscope)

四通道示波器是 Multisim 中新增加的一种仪器,它也是一种用来显示信号波形的形状、大小以及频率等参数的仪器,四通道示波器的图标和面板如图 2.4.9 所示,其使用方法与两通道示波器相似,但存在以下不同点。

(1) 信号输入通道有 A、B、C、D 四个。

(2) 在设置各个通道 Y 轴输入信号的量程时,通过单击“通道选择按钮”来选择要设置的通道。

(3) 按钮  相当于两通道示波器中的 Add 按钮,即 X 轴按设置时间进行扫描,而 Y 轴显示通道 A 与通道 B 的输入信号之和。

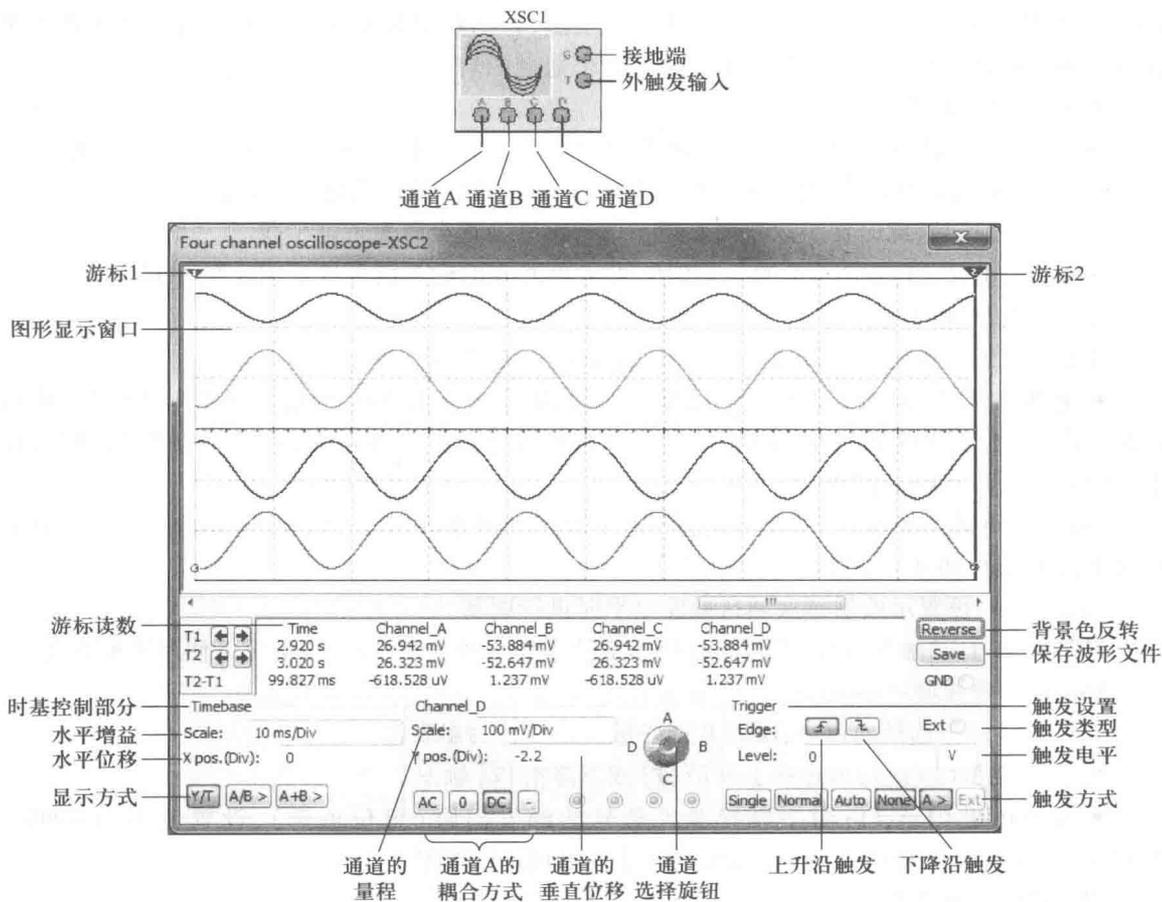


图 2.4.9 模拟四踪示波器的图标和面板

(4) 单击 **A/B >** 按钮和 **A+B >** 按钮后, 出现如图 2.4.10 所示各通道运算方法选项集合。

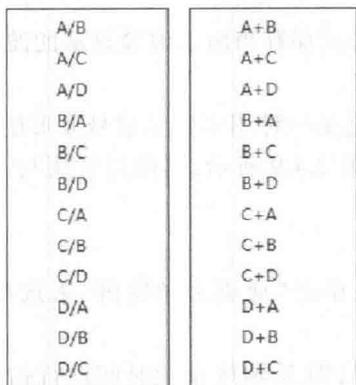


图 2.4.10 各通道运算方法选项集合

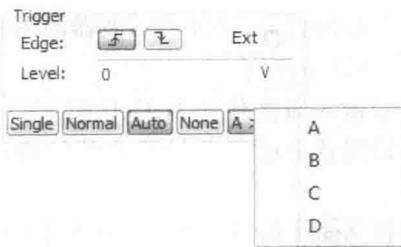


图 2.4.11 内部触发参考通道选择

(5) 单击  按钮,进行内部触发参考通道选择,如图 2.4.11 所示。

6. 波特图仪 (Bode Plotter)

波特图仪可以用来测量和显示电路的幅频特性与相频特性,类似于实验室的扫频仪,还能用于测量信号的电压增益,进行相移特性和频谱分析。波特图仪的图标和面板如图 2.4.12 所示。波特图仪有 IN 和 OUT 两对端口,分别接电路输入端、输出端的正和负端。在使用波特图仪时,必须在电路的输入端接入 AC(交流)信号,对频率没有特殊要求,频率测量的范围由波特图仪的参数设置决定。仿真启动后可以修改波特图仪的参数,如坐标范围及在电路的测试点,但修改后最好重新仿真,以保证曲线的完整与准确。

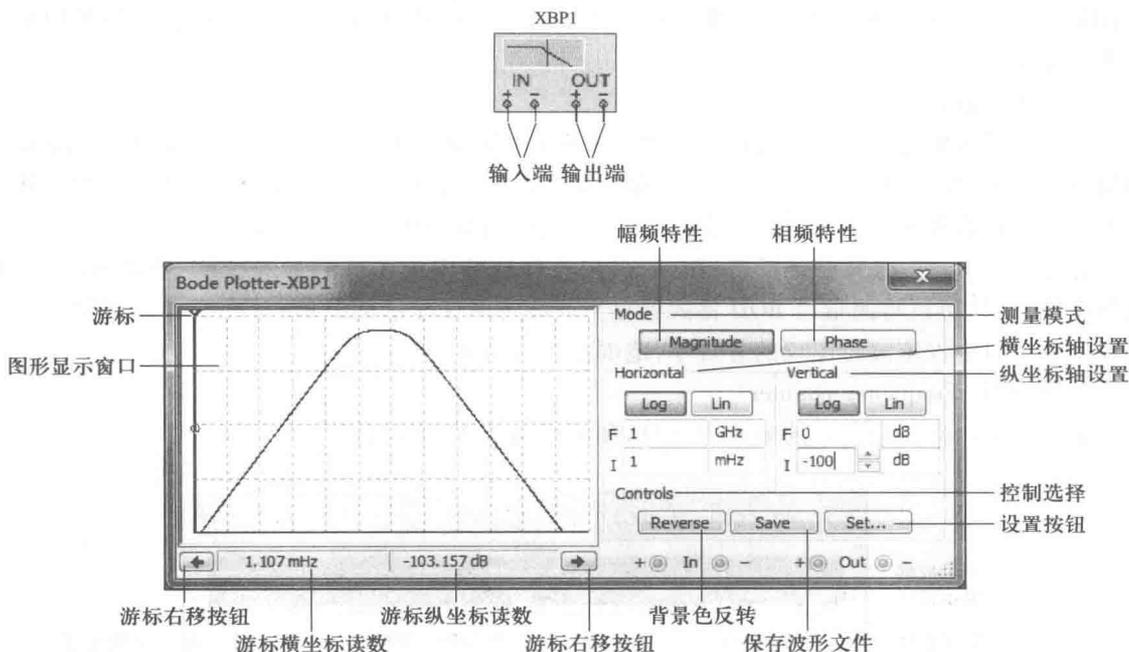


图 2.4.12 波特图仪的图标和面板

(1) 幅频或相频特性分析选择

当单击 Magnitude(幅频特性选择按钮)时,波特图仪测量的是输出/输入电压增益(以 dB 为单位)与频率之间的关系;当单击 Phase(相频特性选择按钮)时,波特图仪测量的是输出/输入的相移大小(以度为单位)与频率之间的关系。

(2) 纵坐标设置 (Vertical)

波特图仪器的纵坐标设置由 3 个部分组成:Log/Lin(对数/线性)坐标选择按钮、坐标的起始值和坐标的终止值。单击 Log 按钮,波特图仪的纵坐标以对数方式显示;单击 Lin 按钮,波特图仪的纵坐标以线性方式显示。纵坐标的初始值和终止值的设置一般要遵循一定的规则,如

表 2.4.1 所示。

表 2.4.1 纵坐标的设置规则

测量模式	显示坐标设置	初始值	终止值
幅频模式	对数坐标	-200 dB	200 dB
幅频模式	线性坐标	0	10e+09
相频模式	线性坐标	-720°	720°

当测量电压增益时,纵坐标指示的是电路输出/输入电压的比值,若使用对数坐标,单位是分贝(dB)。当测量相位时,纵坐标一般指示的是相角,单位是度(°)。需要注意的是,初始值必须小于终止值。

(3) 横坐标设置(Horizontal)

横坐标显示的总是频率,它的对数/线性坐标选择功能与纵坐标的相似。其频率范围取决于初始值(I)和终止值(F)的值。由于频率响应分析需要一个较大的频率范围,所以一般都是用对数坐标。在设置横坐标初始值和终止值时需要注意,初始值要小于终止值。

另外单击 Reverse(背景色反转)按钮,可以改变波特图仪屏幕的背景颜色;单击 Save(保存波形文件)按钮可以将测量以 BOD 格式保存当前波特图仪显示的波形;单击 Set(设置按钮),可在弹出的窗口中设定波特图仪的分辨率,范围是 1~1 000。

7. 频率计(Frequency counter)

频率计是用来测量信号的频率,其图标和面板如图 2.4.13 所示。

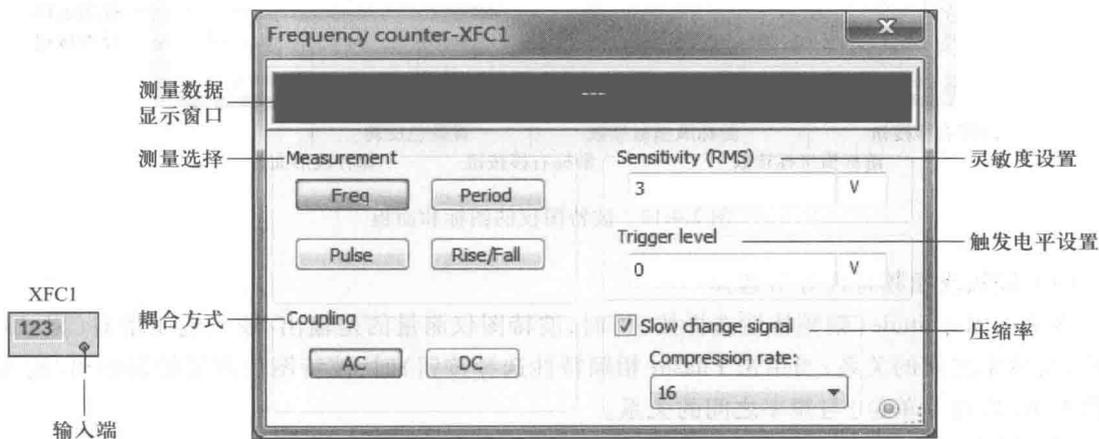


图 2.4.13 频率计的图标和面板

频率计的使用方法如下。

- 测量选择(Measurement): Freq——测量频率; Pulse——测量正、负脉冲持续时间;

Period——测量周期;Rise/Fall——测量单个循环周期的上升沿时间和下降时间。

- 耦合方式(Coupling):AC——显示信号的交流分量;DC——显示信号的直流分量和交流分量之和。

- 灵敏度设置(Sensitivity):改变测量灵敏度,取值为电压的有效值。

- 触发电平设置(Trigger Level):触发电平必须在波形读数之前设置。

以一个简单交流电压电路为例,如图 2.4.14 所示,使用频率计测量交流电路时的不同测量内容结果显示如图 2.4.15 所示。

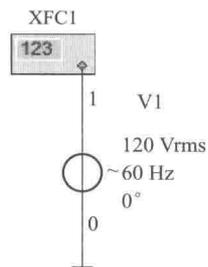
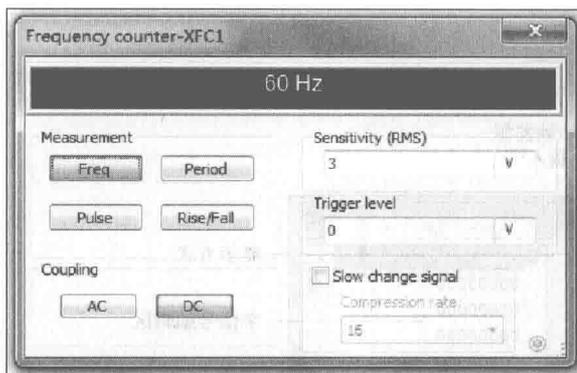
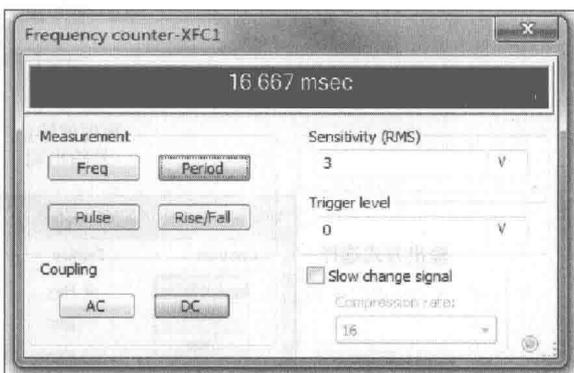


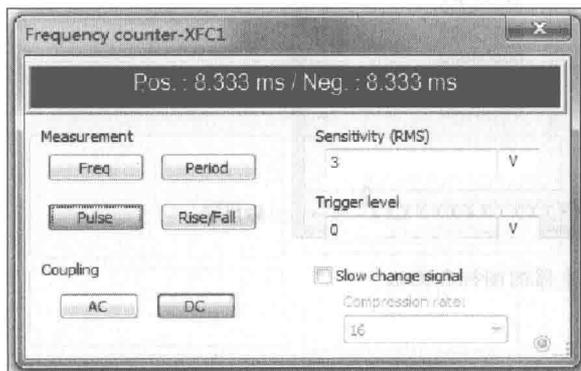
图 2.4.14 示例电路



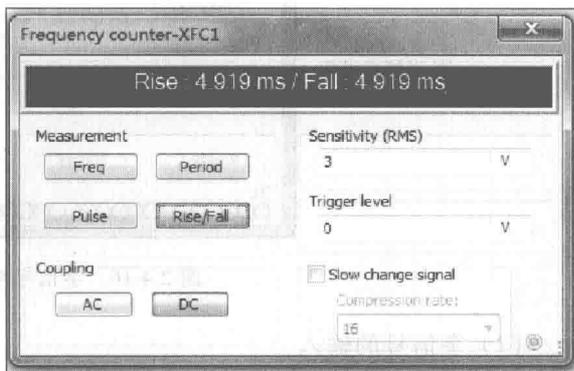
(a) 频率



(b) 相位



(c) 脉冲



(d) 上升/下降时间

图 2.4.15 频率计的测量结果

8. 字信号发生器(Word generator)

字信号发生器是一台能产生 32 位同步逻辑信号的信号源,用于数字逻辑电路进行测试。其图标及面板如图 2.4.16 所示。字信号发生器图标的左右各有 16 个端子,左边为 0~15 端子,右边为 16~31 端子,这 32 个端子是字信号发生器的信号输出端。R 为数据就绪信号输出端,T 为外部触发信号端。

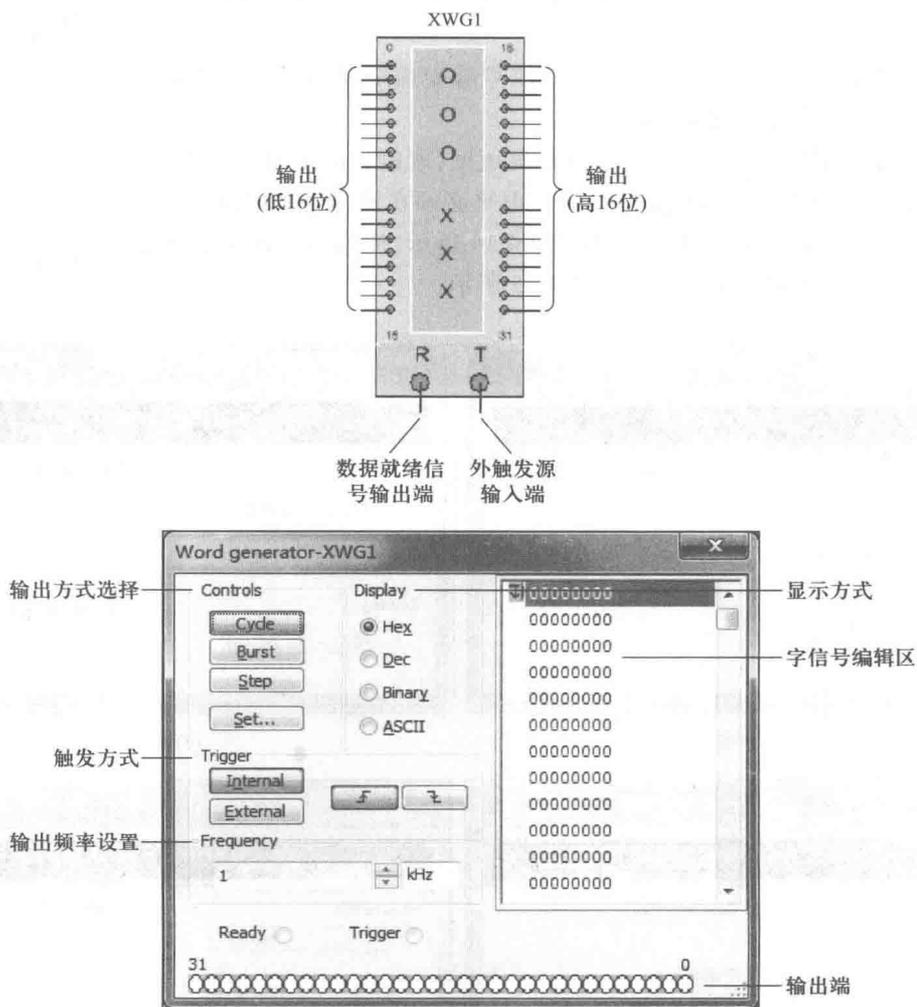


图 2.4.16 字信号发生器的图标和面板

(1) 字信号的输入

在字信号编辑区,32 位字信号以 8 位十六进制数编辑和存放,可以存放 2 560 条字信号,地址编号为 0000~09FF。

字信号输入的操作,是将光标指针移至字信号编辑区的某一行并单击鼠标,由键盘输入如二进制代码的字信号,光标从左到右、自上而下移动,可连续输入字信号。通过字信号显示方式选择,可以编辑或显示不同的字信号格式。

字信号发生器被激活后,字信号将按照一定规律逐行从底部的输出端送出,同时在面板底部对应于输出端的 32 个小圆圈内实时显示输出信号各位的二进制数值。

(2) 字信号的输出方式(Controls)

字信号的输出方式分为循环(Cycle)、单帧(Burst)、单步(Step)3种方式。单击“Step”,将输出一条字信号,这种方式适合于对电路进行单步调试;单击“Burst”按钮,则从首地址到末地址连续逐条地输出一遍字信号;单击“Cycle”按钮,将循环不断地按Burst方式输出字信号。Cycle和Burst方式的输出快慢节奏是由输出频率决定的。

在字信号编辑区,选中某个字信号后,单击鼠标右键可设置指针(Cursor)、断点(Breakpoint)、起始位置(Initial Position)和结束位置(Final Position)等。在使用Burst方式的输出时,运行至断点(Breakpoint)地址输出暂停,再按F9键或仿真暂停开关可恢复运行。

(3) 字信号的触发方式(Trigger)

字信号的触发方式分为内部(Internal)和外部(External)触发方式。当选择“Internal”时,字信号的输出直接由输出方式按钮启动运行;当选择“External”时,需要接入外部触发脉冲信号,并设置上升沿触发或下降沿触发,然后单击输出方式按钮,待触发脉冲到来后启动输出。此外,在数据准备好后,输出端还能得到与输出字信号同步的时钟脉冲输出。

(4) 字信号的保存、调用、清除等操作

单击“Set...”按钮,弹出Setting对话框,如图2.4.17所示。对话框包括打开(Load)、保存(Save)、清除(Clear buffer)、递增(Up counter)、递减(Down counter)、右移(Shift right)、左移(Shift left)等内容,用于对编辑区的字信号进行相应的操作。其中后四个选项用于在编辑区生成按一定规律排列的数字信号,如选择Up counter,则按0000~03FF排列;选择Shift right,则按8000、4000、2000等逐步右移一位的规律排列;其余以此类推。字信号的存盘文件的后缀为“dp”,选择Load,可将已经存在的文件调出使用。

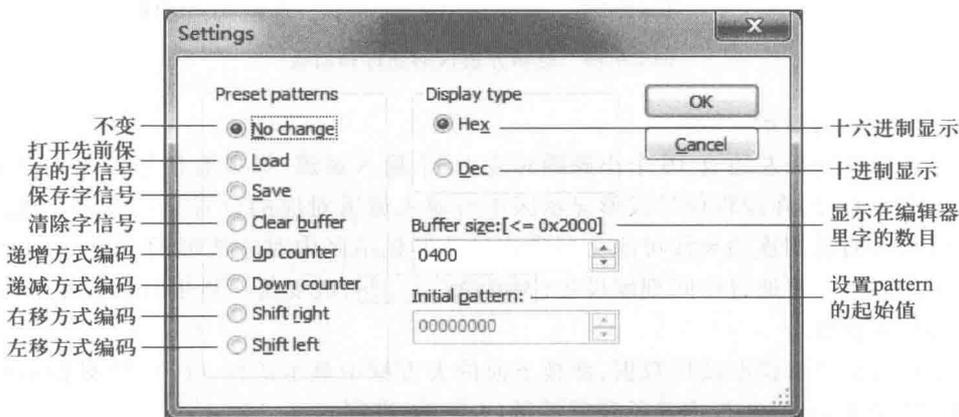


图 2.4.17 字信号发生器的设置对话框

9. 逻辑分析仪(Logic Analyzer)

虚拟逻辑分析仪和实际的仪器相似,可以同步记录和显示16路数字信号。可以用于数字逻

辑信号的高速采集和时序分析,是分析复杂数字信号的有力工具。逻辑分析仪的图标和面板如图 2.4.18 所示。

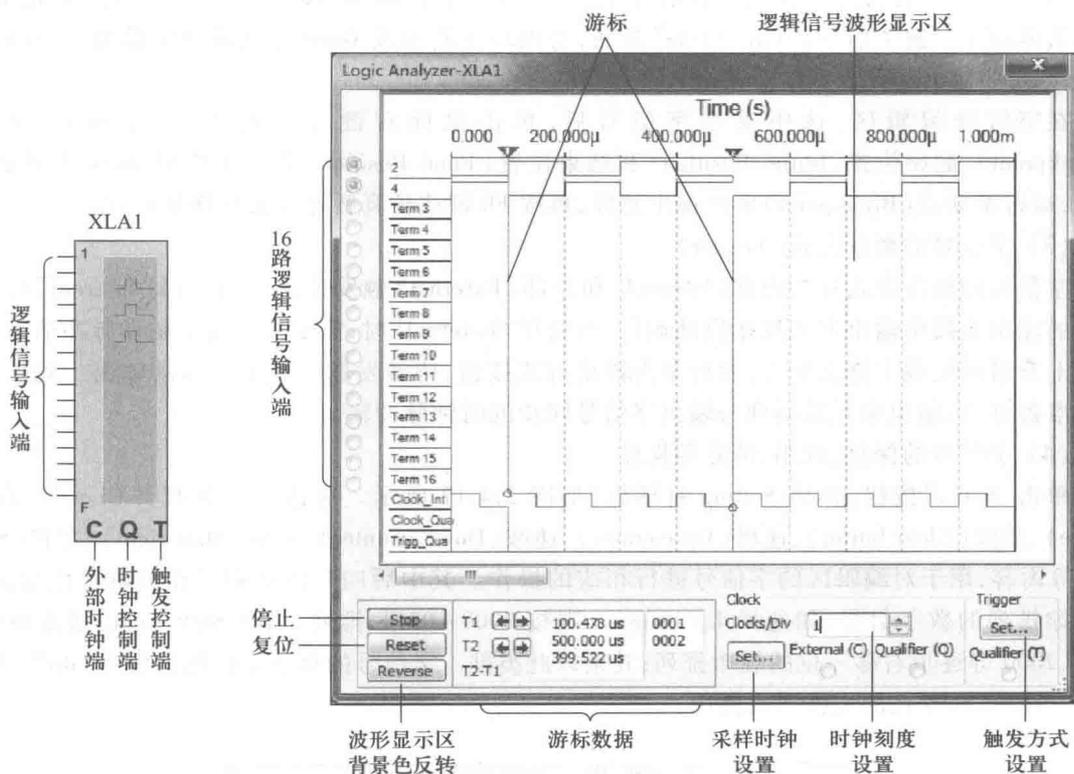


图 2.4.18 逻辑分析仪的图标和面板

(1) 逻辑信号的显示

逻辑分析仪的面板左边有 16 个小圆圈对应 16 个输入通道,如果有信号输入则对应的圆圈内出现一个实心灰点,在逻辑信号波形显示区中与输入通道对应的位置显示波形。通过设置于逻辑分析仪输入通道相连的导线可改变逻辑信号波形显示区中对应波形的颜色,从上到下依次为最低位至最高位。可通过时间刻度设置(**Clocks/Div 1**)改变每一格中时钟脉冲个数。

(2) 游标及读数

通过拖拽游标可以读取波形数据,面板下面的大方框中显示游标 T1 和 T2 处的时间读数及游标 T2 和 T1 的差值,在小框内显示逻辑读数(4 位 16 进制)。

(3) 采样时钟设置(Clock Setup)

单击采样时钟设置区的设置按钮(**Set...**),弹出 Clock Setting 对话框,如图 2.4.19 所示。可以通过以下几个方面对逻辑分析仪的时钟参数进行设置。

- 时钟源设置(Clock source):选择时钟的来源,External 为外部时钟,Internal 为内部时钟。

- 时钟脉冲频率设置 (Clock rate): 对内部时钟的频率进行设置。
- 时钟限制设置 (Clock qualifier): 该设置于外部时钟源配合使用。选择 1 时则输入为 1 时开放时钟; 选择 0 时则输入为 0 时开放时钟; ×代表时钟控制一直开放。
- 采样设置 (Sampling setting): Pre-trigger samples 和 Post-trigger samples 分别用来设置采样前和采样后的显示数据。Threshold voltage.(V) 用来设置门限电压。

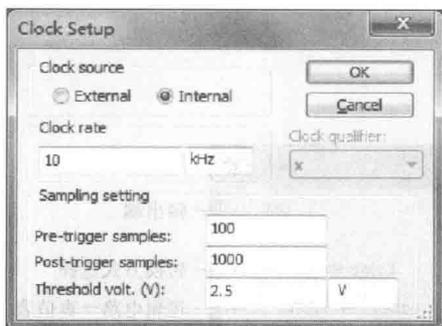


图 2.4.19 采样时钟设置

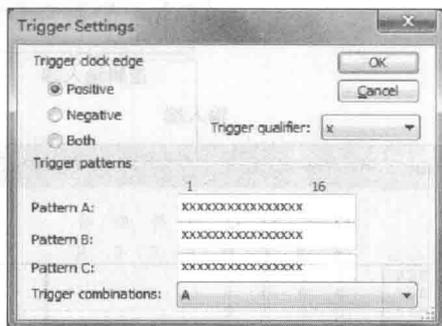


图 2.4.20 触发方式设置

(4) 触发方式设置 (Trigger)

单击触发方式设置区的设置按钮 (Set...), 弹出 Trigger Settings 对话框, 如图 2.4.20 所示。在触发时钟边沿设置 (Trigger clock edge) 里, 可设置 3 种触发方式: 上升沿触发 (Positive)、下降沿触发 (Negative)、上升或下降沿双触发 (Both)。

触发限制设置 (Trigger qualifier) 对触发有控制作用。若 Trigger qualifier 设置为 × 时, 触发控制不起作用, 触发由触发信号决定; 若 Trigger qualifier 设置为 0 (或 1), 则只有在触发信号为 0 (或 1) 时, 逻辑分析仪才触发。

触发方式设置 (Trigger patterns) 有多种选择, 在 Pattern A、Pattern B、Pattern C 中可以设定触发样式, 设置为 × 时代表 0 或 1 都可以。还可以在 Trigger combinations 中设置触发样式组合。单击触发样式组合菜单右边的按钮 (Trigger combinations: A), 在列表中选择 一个组合, 则触发样式被设置为该种组合, 逻辑分析仪在读到一个指定字或几个字的组合后触发。如果 Pattern A、Pattern B、Pattern C 保留默认设置 ××××××××××××××××, 则表示只要第一个输入逻辑信号到达, 无论是什么逻辑状态, 逻辑分析仪均触发, 并开始波形的采样。

10. 逻辑转换器 (Logic Converter)

逻辑转换器是 Multisim 中特有的仪器, 在实际实验室中不存在与此对应的仪器。逻辑转换器能够完成真值表、逻辑表达式和逻辑电路三者之间的相互转换, 这一功能给数字逻辑电路的设计与仿真带来了很大方便。逻辑转换器的图标和面板如图 2.4.21 所示。

逻辑转换器共有 9 个接线柱, 左边的 8 个接线柱与被分析逻辑电路的输入端相连, 最右边的一个接线柱为输出, 与被分析逻辑电路的输出端相连, 如图 2.4.22 所示。图中, 逻辑转换器的输

入端 A、B、C 与逻辑电路的输入相连,而输出与逻辑电路的输出相连。双击逻辑转换器图标,打开面板,可实现以下转换功能。

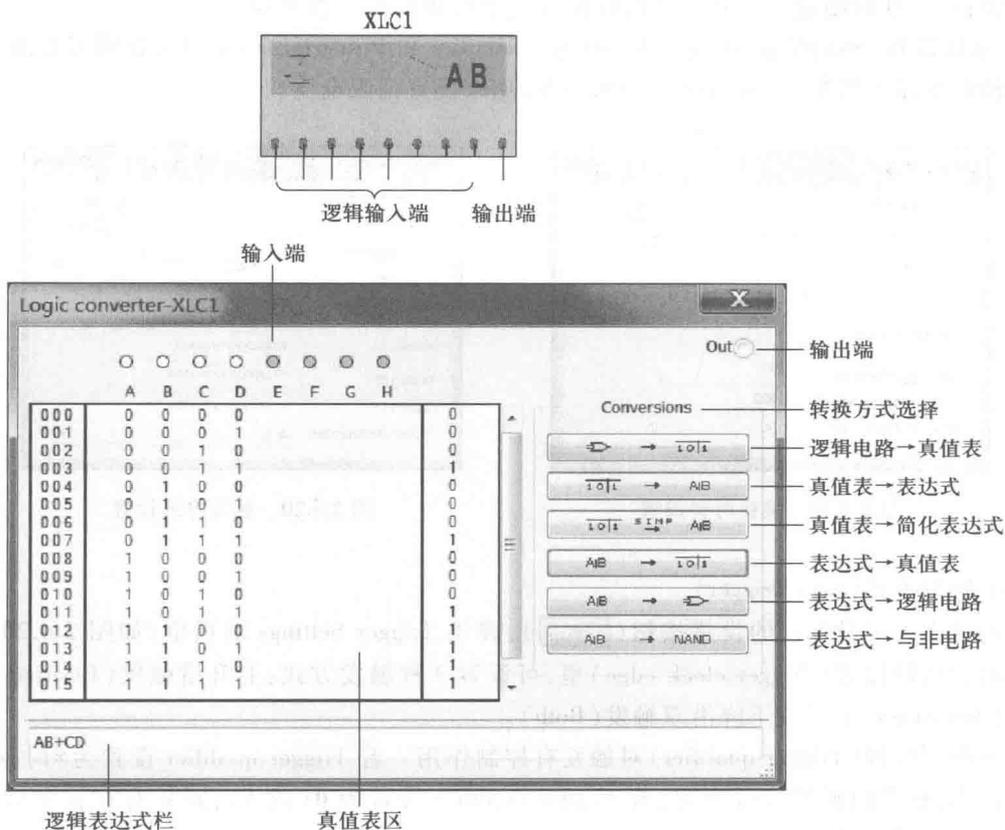


图 2.4.21 逻辑转换器的图标和面板

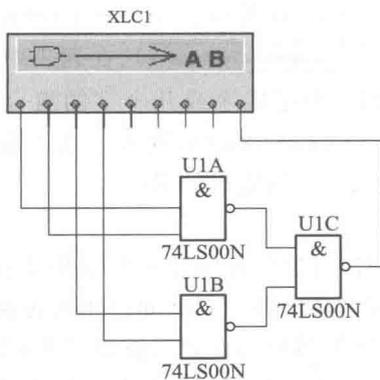


图 2.4.22 用逻辑分析器分析逻辑电路

(1) 逻辑电路→真值表

逻辑转换器有 8 个输入端 A~H 和 1 个输出端 OUT,因此可以得到多输入(最多 8 个输入)单输出的逻辑电路的真值表。首先画出逻辑电路图,将电路输入端连接至逻辑转换器的输入端,电路输出端连接至逻辑转换器的输出端,按下按钮 ,即可在真值表区得到该电路的真值表。

单击逻辑电路到真值表按钮() ,则根据被分析逻辑电路的逻辑关系自动生成一个真值表。

(2) 真值表→逻辑表达式

单击真值表到逻辑表达式按钮() ,可以由真值表导出逻辑表达式,并显示在逻辑表达式栏中。要从真值表导出逻辑表达式,必须在真值表区中输入真值表。输入方式有两种:若已知逻辑电路结构,可用逻辑电路转换为真值表的方式产生;或者直接在真值表栏中输入真值表,根据输入变量的个数单击逻辑转换器面板顶部代表输入端的小圆圈(A~H),选定输入变量。变量被选中后与之对应的小圆圈内部会变白。此时,在真值表栏将自动出现输入变量的所有组合,而右侧靠近滚动条的输出列的初始值全部为“?”。然后根据所要求的逻辑关系来确定或修改真值表的输出值(0、1、×),其方法是多次单击真值表栏右面输出列的输出值,此时便会自动出现 0、1 或×。如想删除新加的变量,则只需再次单击顶部对应变量的小圆圈即可。

(3) 真值表→简化表达式

单击真值表到简化表达式按钮() ,则由真值表导出简化后的逻辑表达式。简化后的逻辑表达式只有与和或两种逻辑关系。其中,逻辑表达式的非用“'”表示,如 \bar{A} 表示为 A'。

(4) 逻辑表达式→真值表

单击逻辑表达式到真值表按钮() ,则根据逻辑表达式栏的表达式生成一张真值表。在输入表达式时,用“'”表示逻辑非。

(5) 逻辑表达式→逻辑电路

单击逻辑表达式到逻辑电路按钮() ,则根据逻辑表达式栏中的表达式生成对应的逻辑电路。

(6) 逻辑表达式→与非门电路

单击逻辑表达式到与非门电路按钮() ,则根据逻辑表达式栏中的表达式生成一个只有与非逻辑关系的(组合)逻辑电路。

11. 伏安特性分析仪(IV Analyzer)

伏安特性分析仪(IV Analyzer)主要用来测量二极管(Diode)、双极型晶体管(BJT NPN 和 BJT PNP)和场效应晶体管(PMOS 和 CMOS)的伏安特性曲线,类似于实验室的晶体管特性测试仪。需要注意的是伏安特性分析仪只能测量未连接到电路中的单个元件。所以在测量电路里的设备之前,可以先将其从电路里断开。

伏安特性分析仪的图标和面板如图 2.4.23 所示,伏安特性分析仪有 3 个接线端,从左至右分别接三极管的 3 个极或二极管的 P、N 结。

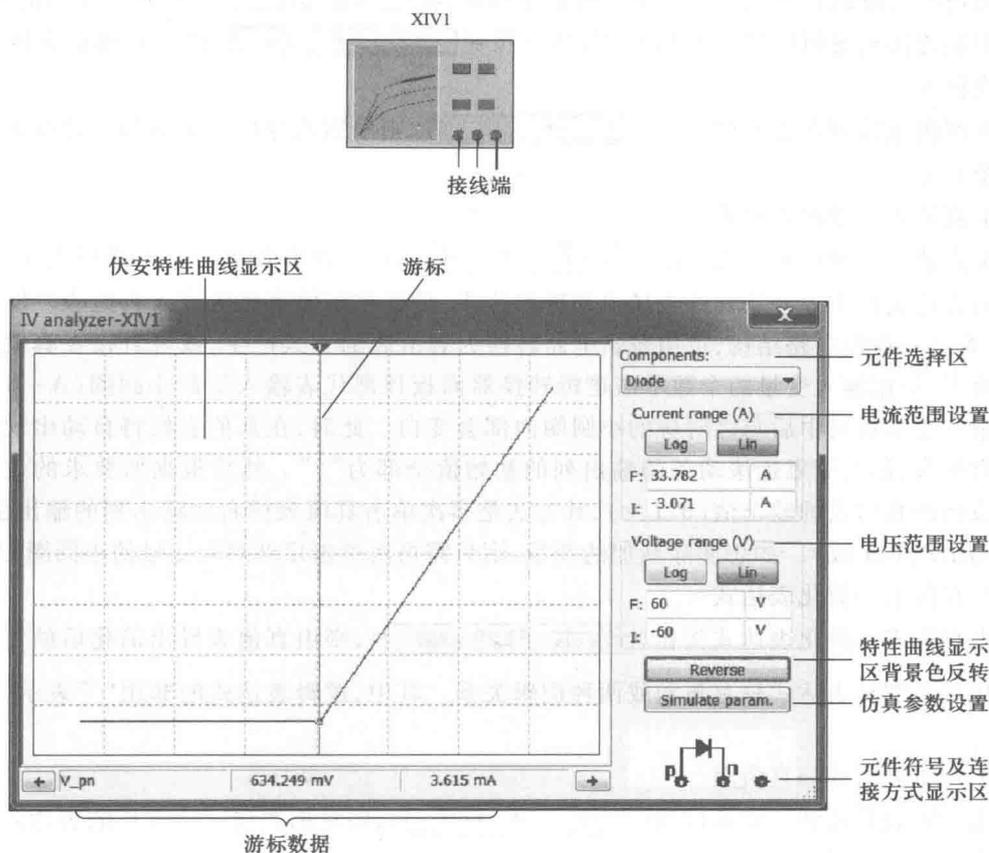


图 2.4.23 伏安特性分析仪的图标和面板

伏安特性分析仪的面板由伏安特性曲线显示区、元件选择区 (Components)、电流范围设置区 (Current range)、电压范围设置区 (Voltage range)、仿真参数设置 (Simulate param.) 以及元件符号及连接方式显示区 6 部分组成,具体如下。

(1) 元件选择区 (Components): 单击元件选择窗口 (右侧下拉箭头后,可选择测试的管子类型,共有 Diode、BJT NPN、BJT PNP、PMOS 和 CMOS 等 5 种。

(2) 电流范围设置区 (Current range (A)): 用于改变伏安特性曲线显示区的电流显示范围。

- F 区 (): 用来设置纵轴电流终止值及其单位。
- I 区 (): 用来设置纵轴电流初识值及其单位。
- Log 按钮 (): 用来设置纵轴对数刻度坐标。

• Lin 按钮():用来设置纵轴线性刻度坐标。

(3) 电压范围设置区(Voltage range(V)):用于改变伏安特性曲线显示区的电压显示范围,其设置方法与电流范围设置类似。

(4) 仿真参数设置(Simulate param.):单击 按钮,弹出如图 2.4.24 所示的 Simulate Parameters 对话框,可以设置元件测试所需的扫描参数。

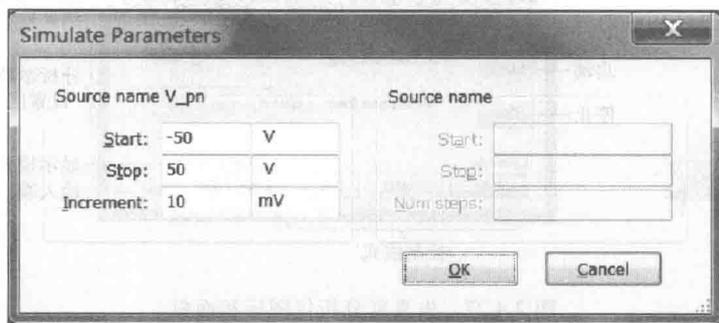


图 2.4.24 Simulate Parameters 设置对话框

以测试增强型 P 沟道 MOS 管为例,测试电路如图 2.4.25 所示,然后合理设置面板参数 Current range(A) 和 Voltage range(V) 以及仿真参数 Simulate param., 点击仿真开关,在伏安特性分析仪的屏幕就会显示该器件的 IV 特性曲线,如图 2.4.26 所示。

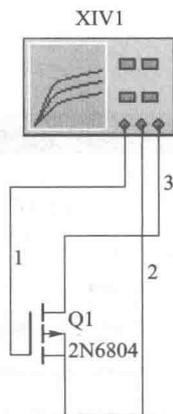


图 2.4.25 测试电路

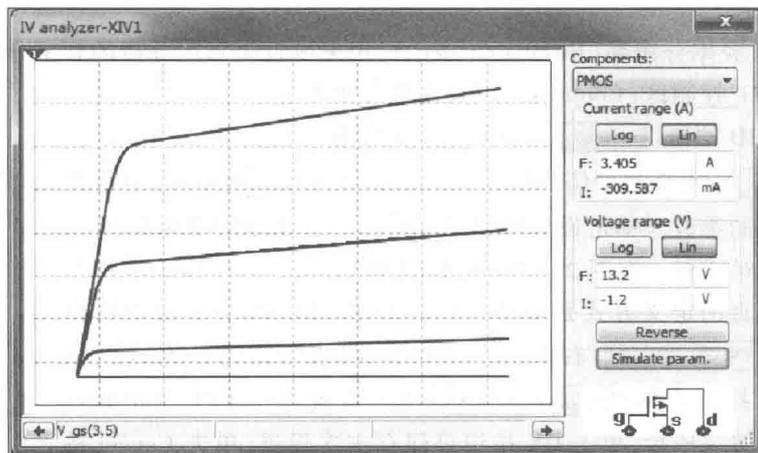


图 2.4.26 2N6804 的 IV 特性曲线

12. 失真度分析仪(Distortion analyzer)

失真度分析仪(Distortion analyzer)是一种测试电路总谐波失真与信噪比的仪器,能够对频率

在 1 Hz~4 GHz 范围内的信号失真度进行测试,包括音频信号。失真度分析仪的图标和面板如图 2.4.27 所示,失真度分析仪只有一个接线端连接被测电路的输出端。失真度分析仪面板中各个部分的功能及设置如下。

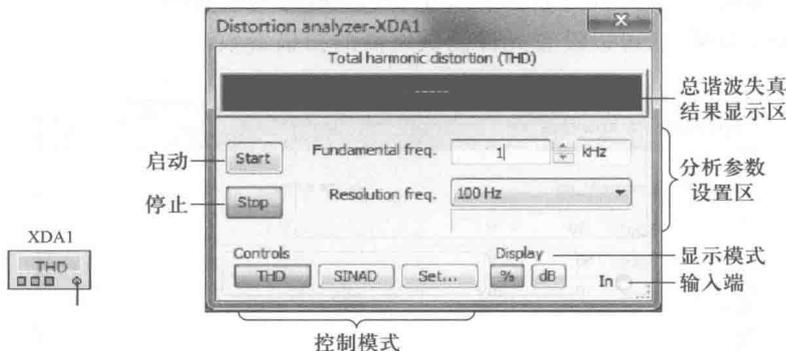


图 2.4.27 失真度分析仪图标和面板

(1) 总谐波失真结果显示区 (Total harmonic distortion (THD)): 用于显示所测电路的总谐波失真数值。数值的显示单位可由显示模式设置 (Display) 中的按钮来改变: 用百分比表示时可单击 **%** 按钮; 用 dB 表示时可单击 **dB** 按钮。

(2) 分析参数设置区: 该区有 2 个选项, 其作用如下。

- 基频 (Fundamental freq.): 用来设置失真度分析的基准频率, 基频值范围可设置为 20 Hz 到 20 kHz。
- 频率分辨率 (Resolution freq.): 用来设置失真度分析的频率分辨率。

(3) 控制区 (Controls): 用来设置分析方法。

THD 按钮 (**THD**): 用于测试总谐波失真。SINAD 按钮 (**SINAD**): 用来测量信号信噪比。Set... 按钮 (**Set...**): 用于设置测试的参数。单击设置按钮 (**Set...**), 弹出如图 2.4.28 所示 Setting 对话框。在图 2.4.28 中, THD Definition 选项用于选择总谐波失真的定义方式是 IEEE 还是 ANSI/IEC; 单击谐波次数窗口 (**Harmonic num. 10**) 右侧的上下箭头, 可设置谐波的次数; FFT 点数下拉列表框 (**FFT points 1024**) 用于设置进行 FFT 变化的点数。设置完毕后, 单击 OK 按钮可保存本次设置, 单击 Cancel 按钮可取消。

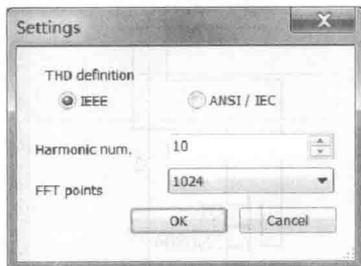


图 2.4.28 测试参数设置

13. 频谱分析仪 (Spectrum analyzer)

频谱分析仪 (Spectrum analyzer) 是一种测试高频电路频域的测量仪器, 主要用来分析电路的幅频特性, 类似于时域的示波器, 能够测量信号的功率和所含的频率成分。频谱分析仪的图标和

面板如图 2.4.29 所示,频谱分析仪有两个接线端,端子 IN 用于连接被测电路的输出端,端子 T 用于连接外部触发信号。

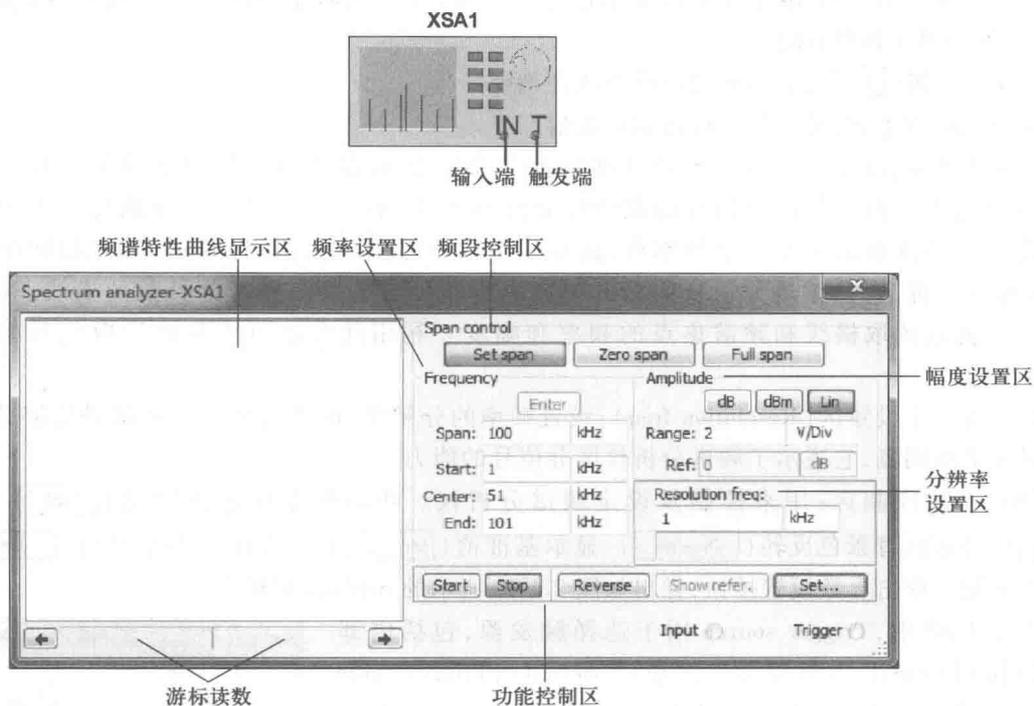


图 2.4.29 频谱分析仪的图标和面板

频谱分析仪面板左侧为频谱特性曲线显示区,右侧又分为四个区,分别是频段控制区(Span control)、频率设置区(Frequency)、幅度设置区(Amplitude)、分辨率设置区(Resolution freq)、功能控制区等。具体功能如下。

(1) 频段控制区(Span control):控制频率变化范围的显示方式。

- Set span 按钮():表示频率由频率设置区(Frequency)设定。
- Zero span 按钮():表示仿真的结果由频率设置区(Frequency)中的 Center 文本框()所设定的频率为中心频率。

- Full Span 按钮():表示频率设定范围为全频段,即 0~4 GHz。

(2) 频率设置区(Frequency):主要用于设置频率范围。

Span():设置频率的变化范围。Start():设置起始频率。Center():设置中心频率。End():设置终止频率。

(3) 幅度设置区(Amplitude):频谱纵坐标刻度设置。

- dB 按钮():表示纵坐标用 dB,即以 $20\log_{10}(V)$ 为刻度。

• dBm 按钮 (dBm): 表示纵坐标用 dBm, 即以 $10\log_{10}(V/0.775)$ 为刻度。0 dBm 是电压为 0.775 V 时, 在 600 Ω 电阻上的功耗, 此时功率为 1 mW。如果一个信号是 +10 dBm, 则意味着其功率为 10 mW。在以 0 dBm 为基础显示信号功率时, 终端电阻是 600 Ω 的应用场合 (如电话线), 直接读 dBm 会很方便。

• Lin 按钮 (Lin): 表示纵坐标使用线性刻度。

• Range 文本框: 设置纵坐标每格的幅值。

• Ref 文本: 设定基准值。所谓基准值就是确定显示窗口中信号频谱的某一幅值所对应的频率范围。由于频谱分析仪的数轴没有标明大小, 通常利用游标来读取每一点的频率和幅度。如果读取的不是一个频率点, 而是一个频率范围, 则需要与 Show refer. 按钮配合使用, 单击该按钮, 则在频谱分析仪的显示窗口中出现以 Ref 所设置的分贝数的横线, 移动游标即可方便地读取横线和频谱焦点的频率和幅度。利用此方法可以快速读取信号频谱的带宽。

(4) 分辨率设置区 (Resolution freq): 设置频率的分辨率, 所谓频率分辨率就是能够分辨频谱的最小谱线间隔, 它表示了频谱分析仪区分信号的能力。

(5) 功能控制区: 用来控制及设定频谱分析仪。功能控制区包括启动 ()、停止 ()、显示区背景色反转 ()、显示基准值 () 以及触发参数设置 () 等 5 个按钮。单击参数设置按钮, 弹出如图 2.4.30 所示的 Settings 对话框。

图 2.4.30 中, Trigger source 用于选择触发源, 包括内部触发按钮 (Internal) 及外部触发按钮 (External); Trigger mode 用于选择触发方式, 包括连续触发按钮 (Continuous) 和单次触发源 (Single); 改变 Threshold volt. (V) 选项中的数值可改变阈值电压, 改变 FFT points 选项可改变傅里叶变换的点数。

此外还需注意, 频谱分析仪是对输入的时域信号进行 Fourier 变换 (FFT), 开始时由于变换样本少, 结果不精确, 仪器波形不稳定。当仪器内部 FFT 变换的频率分辨率等于用户设定的值时, 波形趋于稳定, 这时测量得到的结果较为精确。

14. 网络分析仪 (Network analyzer)

网络分析仪是一种测试二端口网络的仪器, 常常用来分析高频电路中的衰减器、放大器、混频器、功率分配器等电子电路及元件的特性。NI Multisim 12 提供的网络分析仪不仅可以测量二端口网络的 S 、 H 、 Y 和 Z 参数, 还可以测量功率增益、电压增益、阻抗等参数, 另外还能为 RF 电路的匹配网络设计提供帮助。网络分析仪图标和面板如图 2.4.31 所示。

网络分析仪有两个接线端, P1 端用来连接被测电路的输入端口, P2 端用来连接被测电路的输出端口。仿真时, 网络分析仪自动对电路进行两次交流分析, 第一次交流分析用来测量输入端

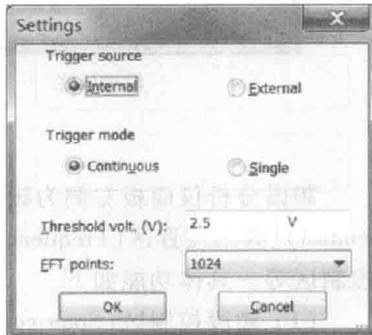


图 2.4.30 频谱分析仪的图标和面板

的前向参数 S_{11} 、 S_{21} ，第二次交流分析用来测量输出端的反向参数 S_{22} 、 S_{12} 。 S 参数确定后，就可以利用网络分析仪以多种方式查看数据，并将这些数据用于进一步的仿真分析。

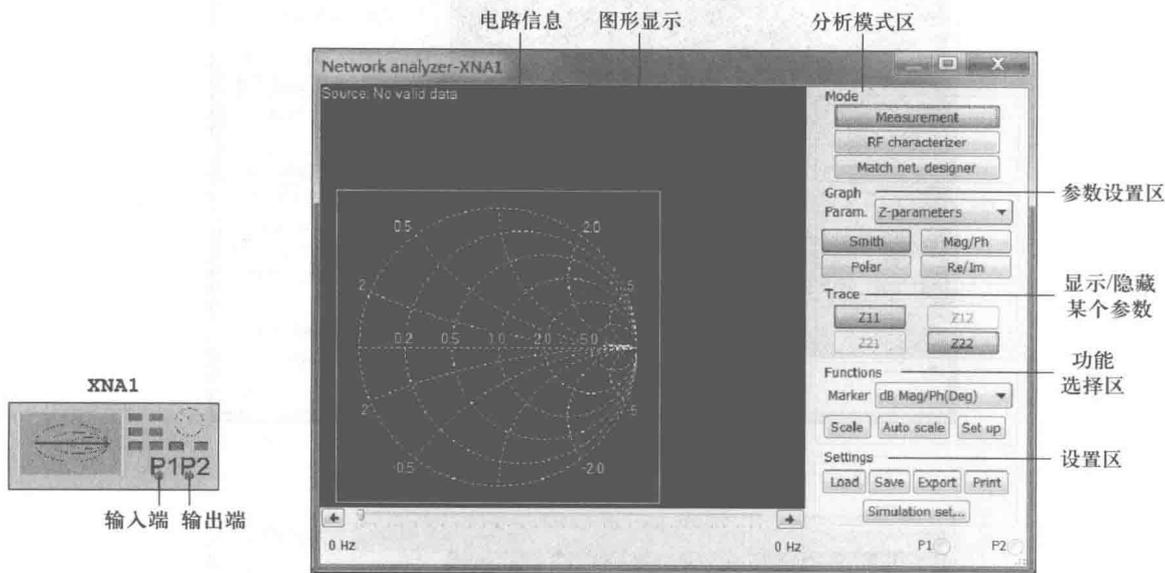


图 2.4.31 网络分析仪的图标和面板

网络分析仪面板的左侧是显示窗口，用于显示电路的 4 种参数、曲线、文本以及相关的电路信息。右侧是 5 个参数设置区域，其具体功能如下。

(1) 分析模式区 (Mode): 设置仿真分析模式，该选择将直接影响其他设置区域的内容。包含按钮测量 (**Measurement**)、射频电路分析 (**RF characterizer**)、高频电路设计 (**Match net. designer**) 等 3 个分析模式选择按钮。

(2) 参数设置区 (Graph): 设置仿真分析参数及其结果显示模式。

- 参数选项 (Param.): 该选项的内容与分析模式选项有关。在 Measurement 模式下有 S 参数 (S -parameters)、 H 参数 (H -parameters)、 Y 参数 (Y -parameters)、 Z 参数 (Z -parameters) 和稳定因素 (Stability factor) 等 5 个选项。在 RF Characterizer 模式下，有功率增益 (Power gains)、电压增益 (Gains) 和阻抗 (Impedance) 等 3 个选项。

- 结果显示模式按钮: 点击 **Smith** 按钮图形以“史密斯格式”显示，如图 2.4.31 所示；点击 **Mag/Ph** 按钮将显示幅频和相频特性曲线，如图 2.4.32 所示；点击 **Polar** 按钮是以极坐标的形式显示图形，如图 2.4.33 所示；**Re/Im** 按钮是以实部和虚部方式显示，如图 2.4.34 所示。

(3) 显示/隐藏某个参数 (Trace): 设置 Graph 区 Param. 下拉列表中所选择参数类型的具体参数。Graph 区 Param. 列表所选择的参数不同，Trace 区所显示的按钮也不同。

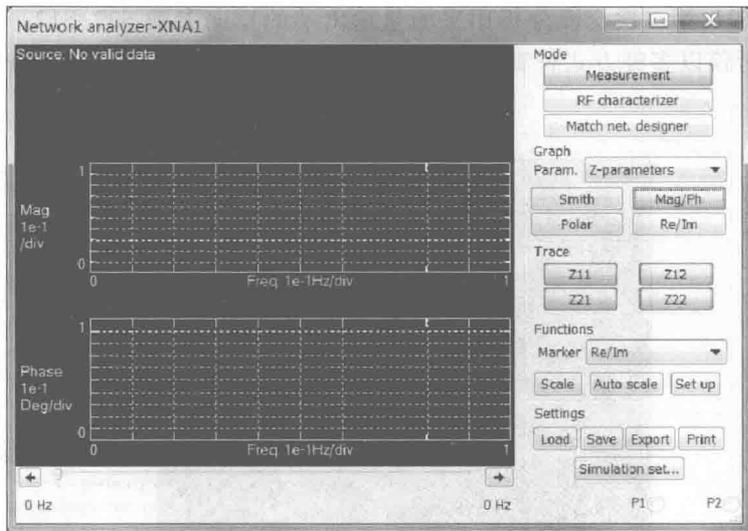


图 2.4.32 频率响应图

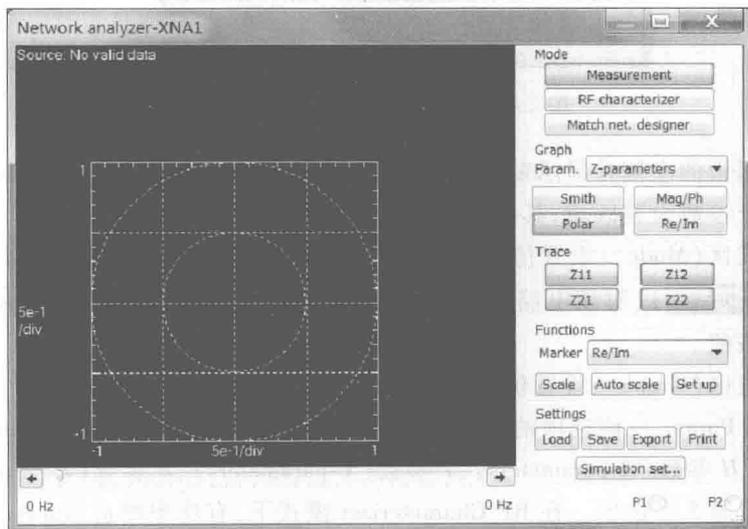


图 2.4.33 极坐标图

(4) 功能选择区 (Functions): 设置仿真分析所需的其他相关参数。

- Marker 下拉列表: 功能是用来设定左边显示区中电路信息的显示模式, 其中包括 3 个选项, 分别为 Re/Im 选项用来设定以直角坐标模式显示参数, Mag/Ph (Deg) 选项用来设定以极坐标模式显示参数, dB Mag/Ph (Deg) 选项用来设定以分贝的极坐标模式显示参数。

- Scale 按钮: 设置纵轴的刻度。只有极点、实部/虚部点和幅度/相位点可以改变。

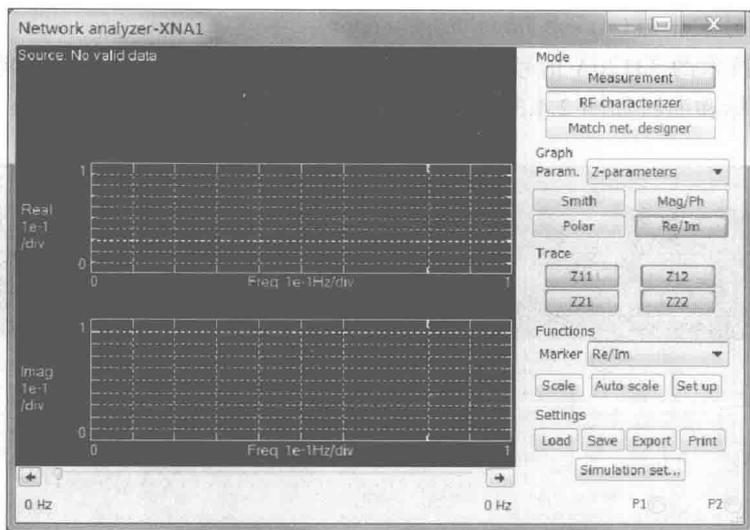


图 2.4.34 实部/虚部显示

- **Auto scale** 按钮:程序自动设置纵轴刻度。
- **Set up** 按钮:用来设定图形的显示模式。单击该按钮弹出 Preferences 对话框,如图 2.4.35 所示,通过该对话框可以设置曲线、网格、绘图区域和文本属性等。

(5) 设置区 (Settings):用来对显示窗口中的数据进行处理。

- **Load** 按钮:加载预先存储的 S-参数数据文件。
- **Save** 按钮:保存当前的 S-参数数据,其扩展名为.sp。
- **Export** 按钮:将数据输出到其他文件。
- **Print** 按钮:打印数据。
- **Simulation set...**:单击该按钮,弹出如图 2.4.36 所示的 Simulation setup 对话框。利用该对话框,可设置仿真的起始频率、终止频率、扫描类型、每十倍坐标刻度的点数和特性阻抗等。



图 2.4.35 图形显示模式设置对话框

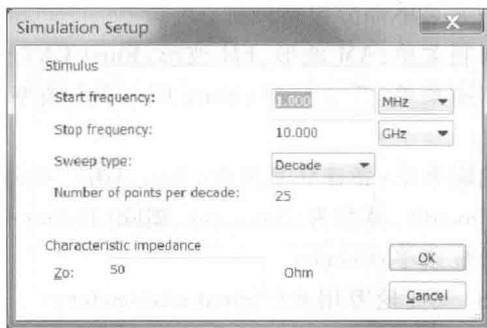


图 2.4.36 测量分析模式设置对话框

15. 安捷伦信号发生器 (Agilent function generator)

基于 Agilent 技术的 33120A 信号发生器是一个能够建立任意波形的高性能的 15 MHz 合成信号发生器,其图标和面板如图 2.4.37 所示。

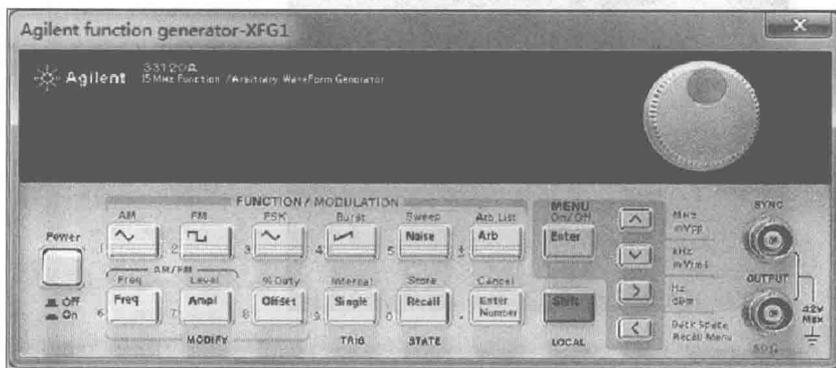


图 2.4.37 安捷伦信号发生器的图标和面板

安捷伦信号发生器支持的功能如下。

(1) 标准波形包括: 正弦 (Sine)、方波 (Square)、三角波 (Triangle)、斜面 (Ramp)、噪声 (Noise)、直流电压 (DC volts)。

(2) 系统任意波形包括: Sinc、负斜面 (Negative Ramp)、升指数 (Exponential Rise)、降指数 (Exponential Fall)、心脏形 (Cardiac)。用户自定义波形为任意类型的 8 到 256 点的波形。

(3) 调制方式有: 无 (NON)、调幅 (AM)、调频 (FM)、Burst、频移键控 (FSK)、Sweep。

(4) 存储部分包括 4 个存储部分, 分别为 #0~#3, #0 为系统默认存储器。

(5) 触发模式包括 Auto/Single, 只适合 Burst 和 Sweep 调制器。

(6) 数据显示屏幕的设置如下。

显示电压: 共采用三个模式, V_{pp} 、 V_{rms} 和 dBm。

编辑数字化数值: 可通过鼠标单击按钮、数字键或者使用旋钮、输入数字键直接输入数值。

(7) 菜单部分包括四种。

调制菜单: AM 波形、FM 波形、Burst CNT、Burst 率、Burst 相位、FSK 频率、FSK RATE。

扫描菜单: 开始频率 (Start F)、停止频率 (Stop F)、扫描时间 (SWP Time)、扫描模式 (SWP Mode)。

编辑菜单: 新建任意波形 (New Arb)、点 (Points)、线形编辑 (Line Edit)、点编辑 (Point Edit)、转换 (Invert)、另存为 (Save as)、删除 (Delete)。

系统菜单: Comma。

16. 安捷伦万用表 (Agilent multimeter)

基于 Agilent 技术的 34401A 万用表是一个 $6\frac{1}{2}$ 位高性能的数字万用表。其图标和面板如

图 2.4.38 所示。

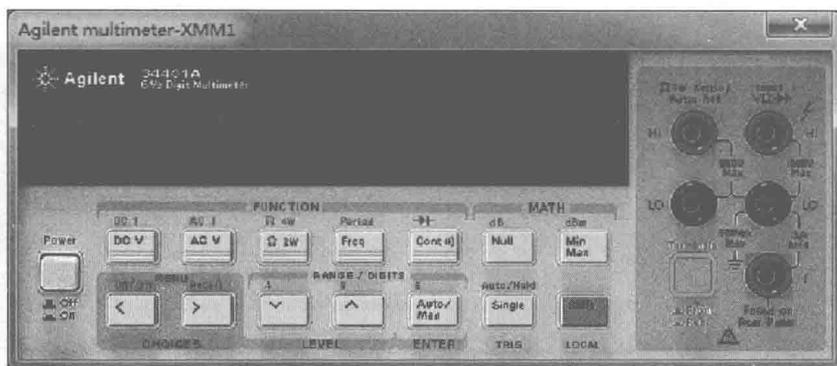
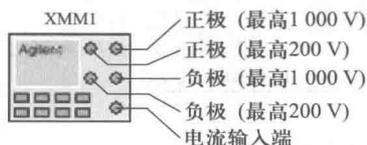


图 2.4.38 安捷伦万用表的图标和面板

安捷伦万用表支持的功能如下。

(1) 测量模式:DC/AC 电压、DC/AC 电流、两导线间的电阻、输入电压信号的频率(范围为 3 Hz~1.999 99 MHz)、输入电压信号的周期、连续性测试、二极管测试、比率测试。

(2) 功能:无(相关测量)、存储的最小-最大可读内容、dB(电压值显示)、dBm(电压值显示)、限制测试(测试时设置最大、最小门限值)。

(3) 触发模式:自动/人工。

(4) 显示模式:自动/人工。

(5) 显示数字: $4\frac{1}{2}$ ~ $6\frac{1}{2}$ 。

(6) 工作菜单包括四种。

测量菜单:连续性、Ratio、数学菜单。

数学菜单:最小-最大、无数值、dB REL、dBm REF R、门限值测试、高门限、低门限。

触发菜单:Read Hold、触发延时。

系统菜单:RDGS 存储、已存 RDGS、蜂鸣、命令。

17. 安捷伦示波器(Agilent oscilloscope)

基于 Agilent 技术的 54622D 示波器是一个 2 通道+16 逻辑通道、100 MHz 带宽的高性能示波器,其图标和面板如图 2.4.39 所示。

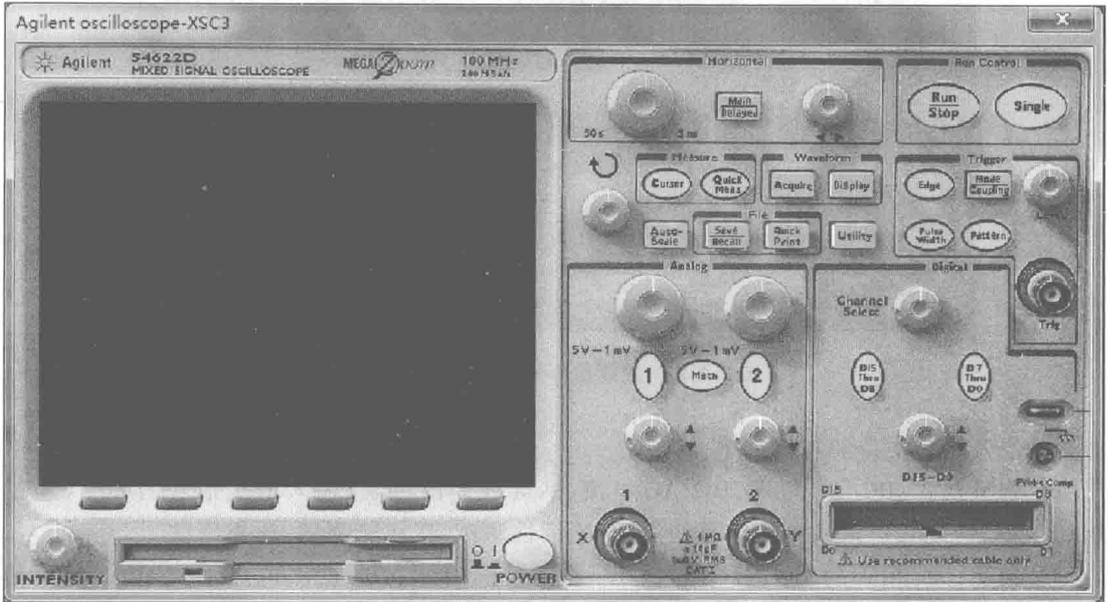
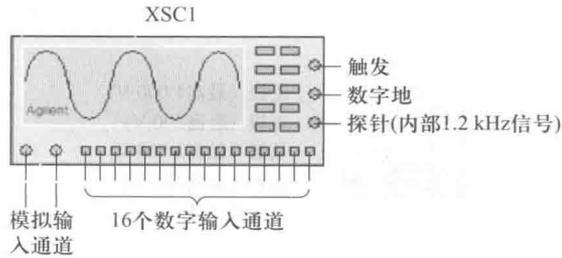


图 2.4.39 安捷伦混合信号示波器的图标和面板

安捷伦示波器面板按钮介绍如下。

POWER: 电源开关。

INTENSITY: 灰度调节旋钮。

POWER 和 INTENSITY 的中间部分为软驱, 软驱上方的一排按钮为设置参数的软按钮, 按钮上方为示波器的显示屏。Horizontal 区为时基调整区, Run Control 为运行控制区, Trigger 区为触发区, Digital 区为数字通道调整区, Analog 区为模拟通道调整区, Measure 区为测量控制区, Waveform 区为波形调整区。

安捷伦示波器的主要功能介绍如下。

- (1) 运行模式: 自动 (Auto)、Single、Stop。
- (2) 触发模式: Auto、Normal、Auto-level。
- (3) 触发类型: 边沿触发、脉冲触发、模式触发。

- (4) 触发源:模拟信号、数字信号、外部触发信号。
- (5) 显示模式:主模式、延时模式、滚动模式、XY 轴模式。
- (6) 信号通道:2 个模拟通道、1 个数学通道、16 个数字通道、1 个用于测试的探针信号。
- (7) 光标:4 个光标。
- (8) 数学通道:傅里叶变换(FFT)、相乘、相除、微分、积分。
- (9) 测量:光标信息、采样信息、频率、周期、峰-峰值、最大值、最小值、上升时间、下降时间、占空比、有效值(RMS)、宽、平均值等。
- (10) 显示控制:向量/点形轨迹(Vector/Point on trace)、轨迹宽、背景色、面板色、栅格色、光标色。
- (11) Auto-scale/Undo:是。
- (12) 打印轨迹图:是。
- (13) 文件操作:将数据保存为 DAT 格式文件,可以转换并显示在系统图形窗口。

18. 泰克示波器(Tektronix oscilloscope)

泰克示波器 TDS 2024 是一个 4 通道、200 MHz 的示波器,其图标和面板如图 2.4.40 所示。



图 2.4.40 泰克数字存储示波器的图标和面板

泰克示波器面板按钮介绍如下。

Run/Stop: 开始或停止对多个触发信号的采样。

Single Seq: 对单个触发信号采样。

Trig View: 查看电流触发信号和触发水平。

Force Trig(强制触发): 立即开始触发信号。

Set to 50%: 将触发水平改变到触发信号的平均值。

Set to Zero: 将时间偏置位置设置为 0。

Help: 进入仪器仪表的帮助界面。

Print: 将图形图表送入打印机打印。

Soft Menu: 支持如下对应的 11 种功能。

① Save/Racall MENU, 保存或重置菜单; ② Measure MENU, 测量菜单; ③ Acquire MENU, 数据采集菜单; ④ Auto Set MENU, 自动设置菜单; ⑤ Utility MENU, 通用程序设置菜单; ⑥ Cursor MENU, 光标设置菜单; ⑦ Display MENU, 显示设置菜单; ⑧ Default MENU, 默认启动设置菜单; ⑨ Channel MENU, 通道设置菜单; ⑩ Math channel MENU, 数学引导菜单; ⑪ Horizontal MENU, 水平设置菜单。

泰克示波器的主要功能介绍如下。

(1) 运行模式: 自动(Auto)、Single、Stop。

(2) 触发模式: Auto、Normal。

(3) 触发类型: 边沿触发、脉冲触发。

(4) 触发源: 模拟信号、外部触发信号。

(5) 信号通道: 4 个模拟通道、1 个数学通道、1 个用于测试的 1 kHz 的探针信号。

(6) 光标: 4 个光标。

(7) 测量内容: 光标信息、频率、周期、峰-峰值、最大值、最小值、上升时间、下降时间、有效值(RMS)、平均值。

(8) 显示控制: 向量/点、颜色对比控制。

19. 探针

探针是 NI Multisim 所提供的一类极具特色的测量工具,它能够方便、快速地检查电路中不同支路、节点或引脚的电压、电流及频率。NI Multisim 12 的仪表工具栏中有测量探针( Measurement probe)和电流探针( Current probe)两种,其中测量探针又可分为动态探针和静态探针两种。在电路仿真时,测量探针可以用来对电路的某个点的电位或某条直流电流,以及频率等电特性进行动态测试。

(1) 动态探针

动态探针只有在仿真运行时有效。在仿真过程中,单击仪表栏中测量探针按钮,测量探针将随着鼠标移动,移动鼠标到被测点,即可得到如图 2.4.41 所示的探针读数标签。若要放弃激活探针,只需再次单击测量探针按钮或按下 Esc 键即可。

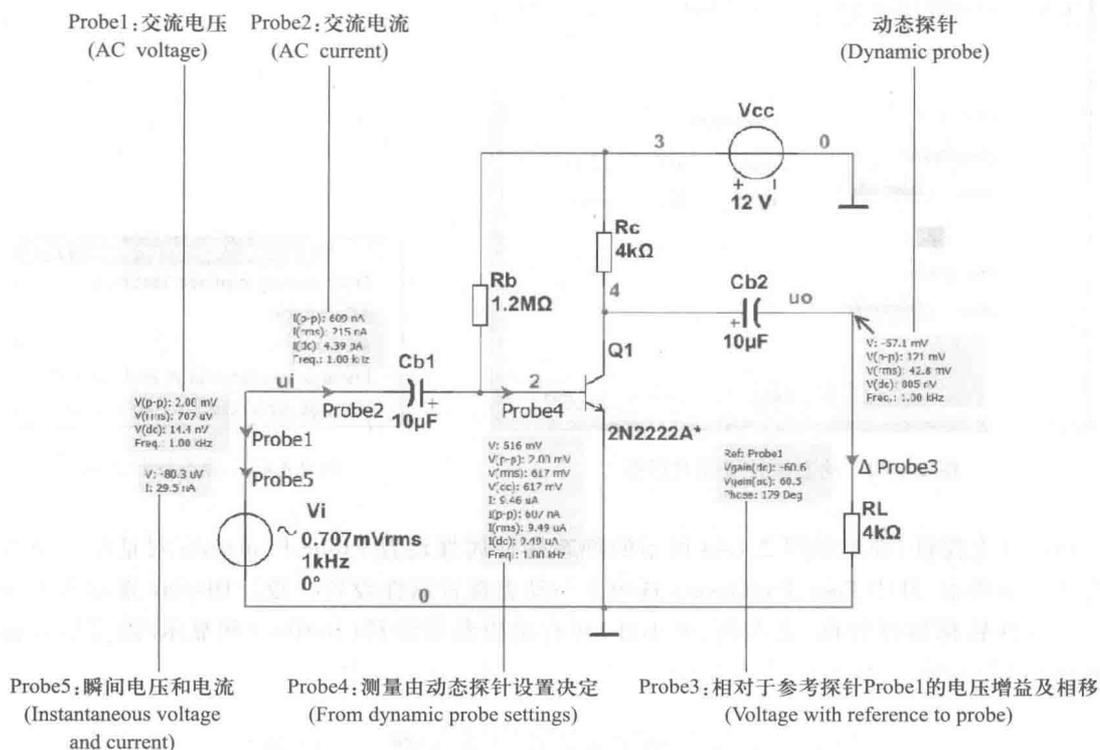


图 2.4.41 动态探针及静态探针读数标签

对动态探针属性进行设置的方法为:单击 Simulate→Dynamic probe properties 菜单,弹出如图 2.4.42 所示对话框。该对话框具有如下功能。

- 显示 (Display): 设置测量探针显示的背景色、文本色和大小。
- 字体 (Font): 设置显示字体的字型和字号。
- 参数 (Parameters): 设置测量参数是否显示,以及显示精度、最大值和最小值。

(2) 静态探针

将鼠标放在测量探针按钮上,弹出静态探针选择菜单,如图 2.4.43 所示的菜单。菜单包含: From dynamic probe settings 读数标签内容由动态探针设置决定; AC voltage 显示交流电压; AC current 显示交流电流; Instantaneous voltage and current 显示瞬态电压和电流; Voltage with reference to probe 读数标签显示探针放置处相对于参考探针的直流电压增益、交流电压增益及相移。

在仿真运行前及运行中,可以将若干个探针放置在电路中需要的节点上,这些探针保持固定,并且包含着来自仿真的数据,如图 2.4.41 所示,直到开始运行另一个仿真或者数据被清除时为止。若要隐藏探针内容,可在探针上单击鼠标右键,在弹出的快捷菜单中选择 Show content 命令,此时探针将仅显示为一个箭头。

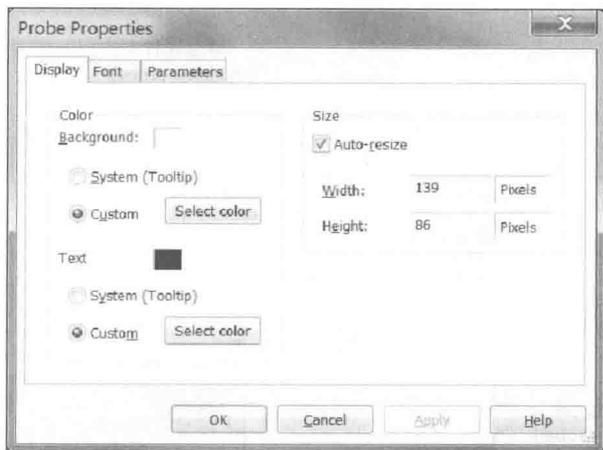


图 2.4.42 动态探针属性对话框

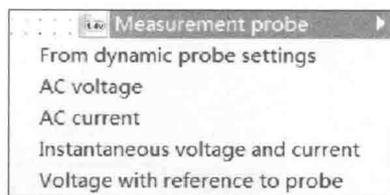


图 2.4.43 静态探针选项

双击静态探针,弹出如图 2.4.44 所示的静态探针属性设置 Probe Properties 对话框。该对话框有 4 个选项卡,其中 Font、Parameters 选项卡与动态探针属性设置一致。Display 选项卡中除了可以设置探针标签背景色、文本色、大小外,还有编辑参考注释(RefDes)和显示/隐藏探针标签(Visibility)等功能。Triggers 选项卡可以为静态标签设置显示条件。



图 2.4.44 静态探针属性设置(Probe Properties)对话框

需要注意的是,在仿真运行时,无论动态探针还是静态探针均不能用来测量电流。

20. Lab VIEW 采样仪器

Lab VIEW 采样器的图标及下拉菜单如图 2.4.45 所示。

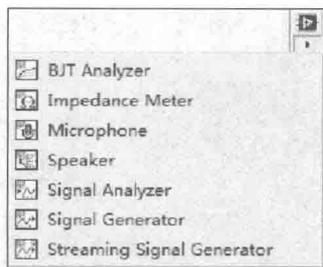


图 2.4.45 Lab VIEW 采样仪器图标及下拉菜单

Lab VIEW 采样仪器下拉菜单所对应的各种采样仪器的图标如图 2.4.46 所示。

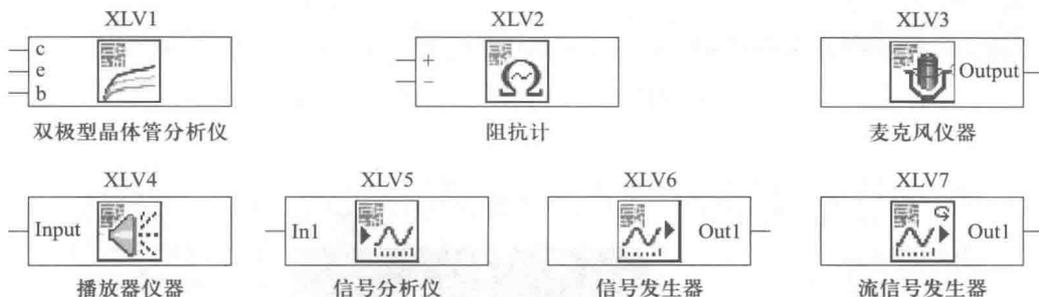


图 2.4.46 Lab VIEW 各种采样仪器的图标

(1) BJT Analyzer 分析仪的使用方法

将 BJT Analyzer 图标接入 NPN 型晶体管,如图 2.4.47 所示。双击 BJT Analyzer 图标,打开 BJT Analyzer 面板,对 Device Type、V_CE Sweep、I_B Sweep 以及图形显示等参数进行合理设置,如图 2.4.48 所示;点击仿真开关,即显示出此类型晶体管的输出特性曲线,如图 2.4.49 所示。

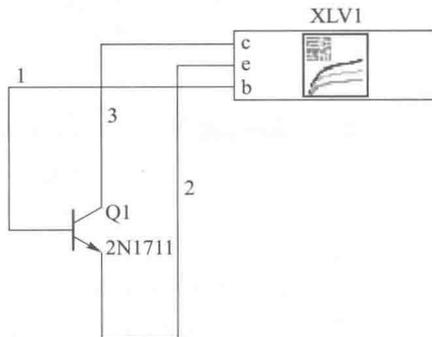


图 2.4.47 BJT Analyzer 的接入

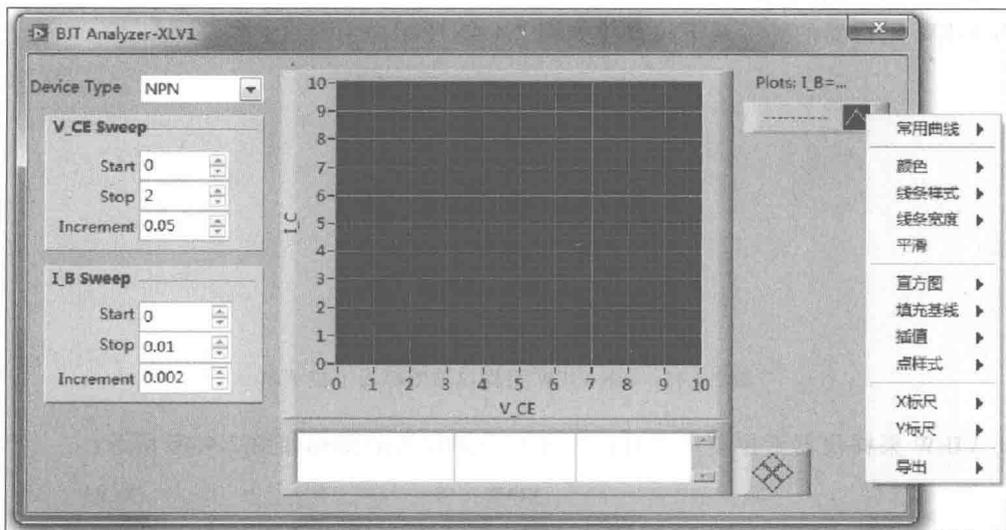


图 2.4.48 BJT Analyzer 的面板及参数设置对话框

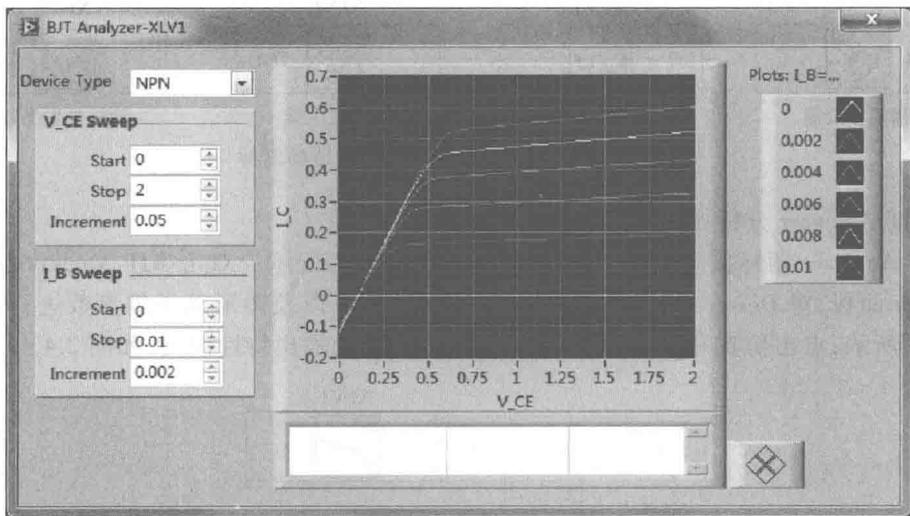


图 2.4.49 BJT Analyzer 的仿真结果

(2) Microphone(麦克风)仪器的使用方法

Microphone 是借助于计算机中的音频输入设备(例如麦克风、CD 播放器等)录制音频数据。Multisim 将作为信号输出这些数据。在开始仿真前配置设置和录制声音,仿真 Multisim 使用该音频信号作为信号源。麦克风仪器的操作步骤如下。

① 将 Microphone 图标放置在原理图编辑区的合适位置,双击打开 Microphone 面板,如图 2.4.50所示。

② 选择所需音频设备(通常使用默认设备)、录音时长和理想抽样率。抽样率越高,输出信号音质越好,但会影响仿真速度。

③ 单击录制声音信号。将与计算机的声音设备相连。

④ 开始仿真前,可以选择重复已录制声音。如果没有选择该选项就仿真电路,一旦仿真时间超过录制信号时长,Multisim 将连续不断地仿真,但是麦克风仪器中的输出信号将降为 0 V。如果选择该选项,麦克风仪器将重复输出已录制数据,直至仿真停止。

⑤ 开始仿真,麦克风仪器将以电压形式输出录制声音。

(3) Speaker(播放器)仪器的使用

Speaker 是借助于计算机的声音设备输出电压信号作为音频信号。在开始仿真前配置设置,并在仿真停止后播放音频。播放器仪器的操作步骤如下。

① 将 Speaker 图标放置在原理图编辑区的合适位置,双击打开 Speaker 面板,如图 2.4.51所示。

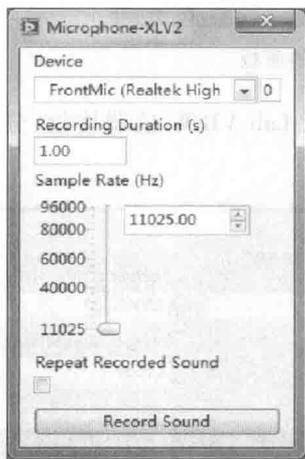


图 2.4.50 Microphone 的面板

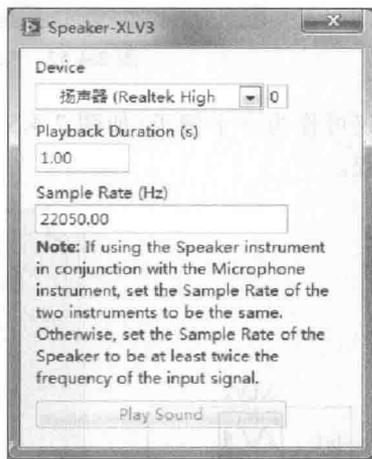


图 2.4.51 Speaker 的面板

② 选择所需音频设备(通常使用默认设备)。如果同时使用麦克风和播放器,两个仪器的抽样率应该相同,否则,播放器的抽样率至少是输入信号频率的两倍。

③ 开始仿真。仿真运行时,播放器仪器采集输入数据,直到仿真时间到达所设置的播放持续范围。

④ 停止仿真,单击 Play,播放仿真中存储的语音。

(4) Signal Analyzer(信号分析仪)的使用

Signal Analyzer 能够显示时域数据、自动功率谱或运行输入信号平均值。Signal Analyzer 的

面板如图 2.4.52 所示。在配置设置参数时,采样率越高,分析效果越好,一般情况下,信号分析仪的采样率为输入信号频率的两倍。

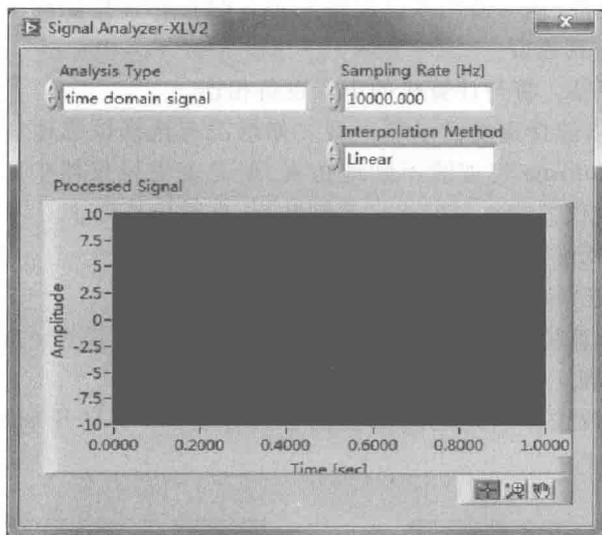


图 2.4.52 Signal Analyzer 的面板

信号分析仪可作为一个例子,如图 2.4.53 所示,显示 Lab VIEW 仪器接收、分析及显示仿真数据的实现过程。

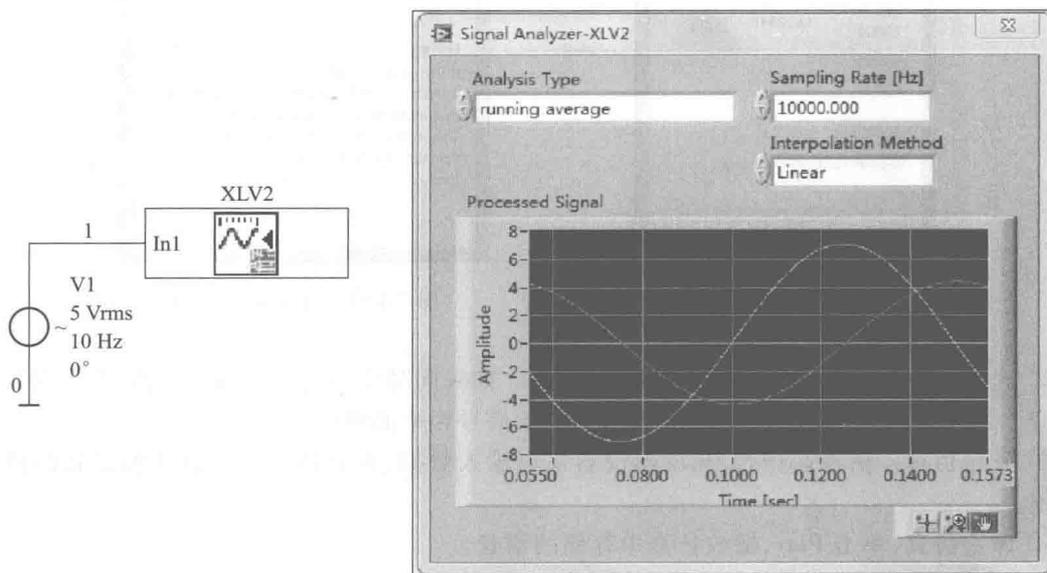


图 2.4.53 实例分析结果

(5) Signal Generator(信号发生器)的使用

Signal Generator 能够产生并输出正弦波、三角波、方波和锯齿波信号。Signal Generator 的面板如图 2.4.54 所示。

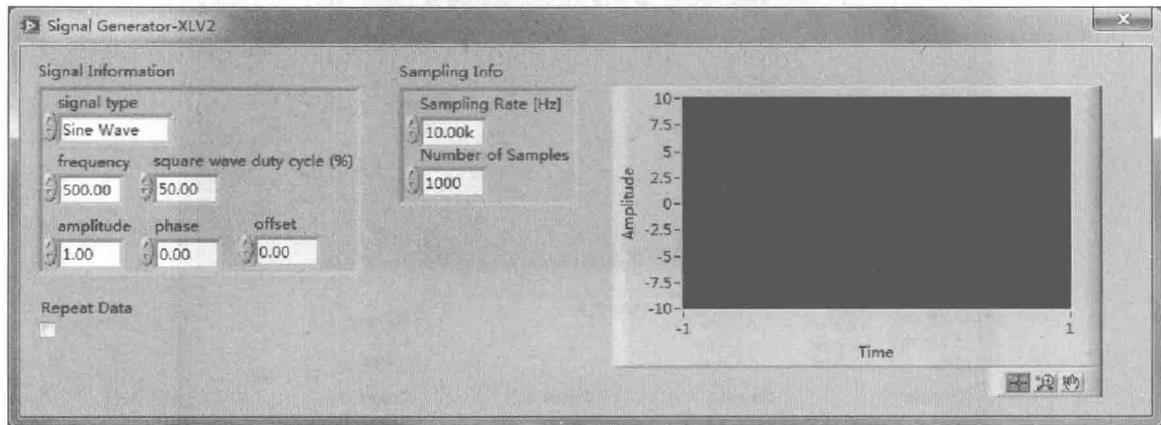


图 2.4.54 Signal Generator 的面板

信号发生器可作为一个简单的 Lab VIEW 仪器的演示例子,可以看到 Lab VIEW 仪器是如何产生或获取数据,然后作为信号源输出进行仿真的过程。图 2.4.55 所示为 Signal Generator 输出频率为 1 kHz、幅值为 1 V 的正弦信号的仿真结果,图 2.4.56 所示为示波器显示 Signal Generator 的输出波形。

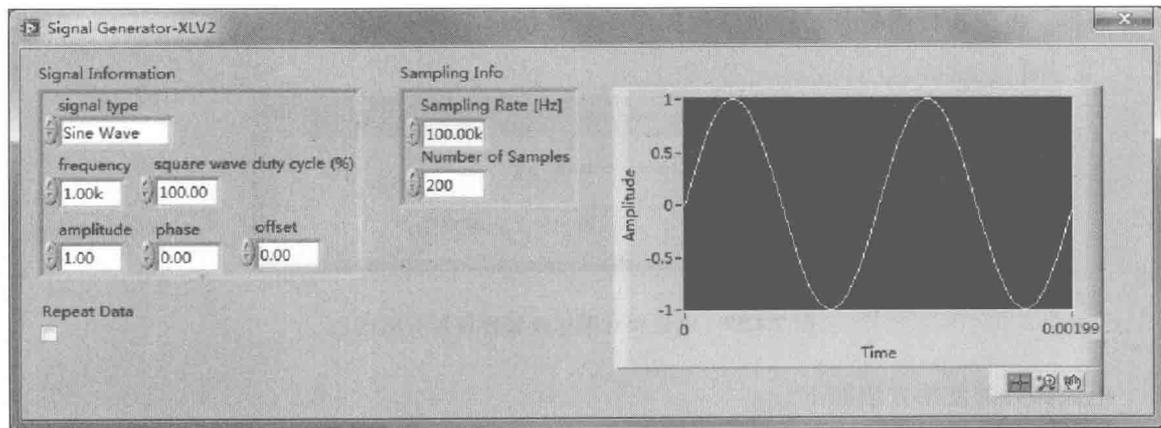


图 2.4.55 Signal Generator 产生的正弦信号

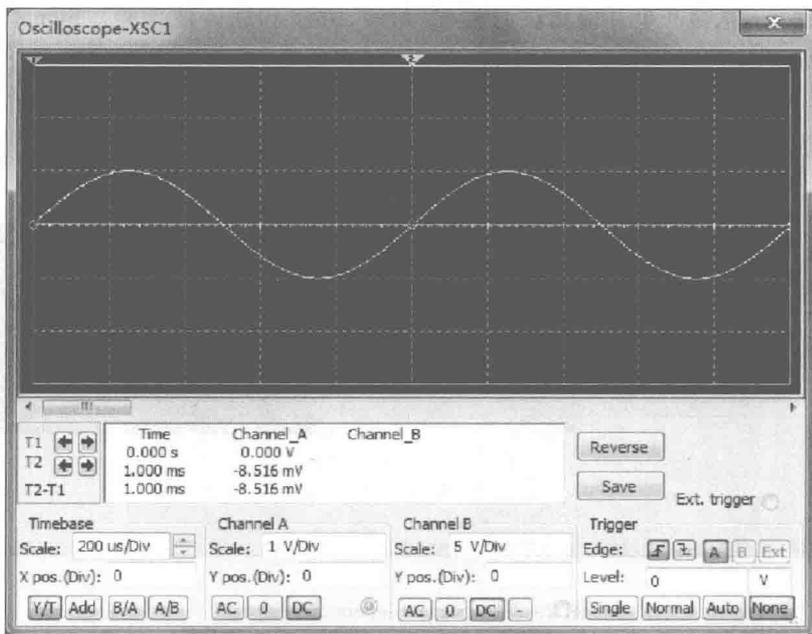


图 2.4.56 示波器显示 Signal Generator 的输出波形

21. 电流探针

电流探针模拟的是能够将流过导线的电流转换成设备输出终端电压的工业用钳式电流探针。电流探针的输出终端与示波器相连,其电流大小由示波器读数及探针的电压-电流转化比计算而得。电流探针图标及属性对话框如图 2.4.57 所示。

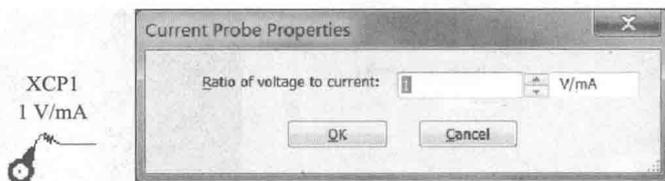


图 2.4.57 电流探针图标及属性设置对话框

电流探针的使用方法如下。

将电流探针图标放到所需测试的电路节点上,如图 2.4.58 所示;双击电流探针图标,打开其设置对话框进行相应的设置,如图 2.4.57 所示;将示波器介入电流探针图标,点击仿真开关,在示波器上得到所需的电流波形,如图 2.4.59 所示。

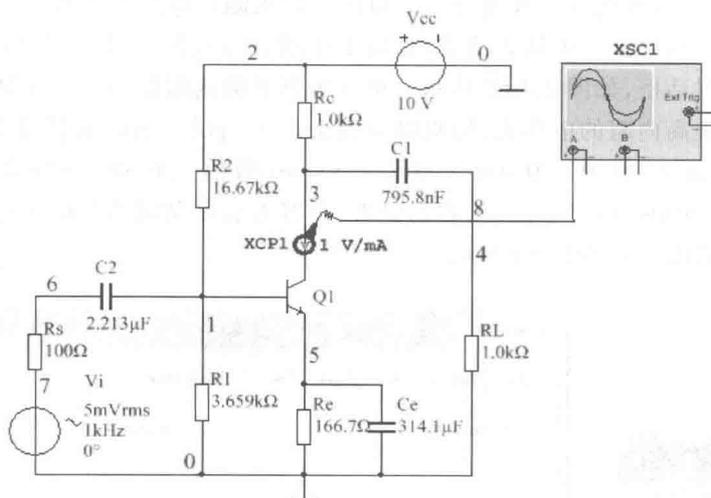


图 2.4.58 电流探针的接入



图 2.4.59 电流波形

22. 电压表

电压表存放在指示元器件库中,在使用中数量没有限制,图标和特性对话框如图 2.4.60 所

示。单击旋转按钮,可以改变其引线的方向。电压表用来测量电路两个节点间的电压。测量时,将电压表并联接在被测电路。电压表的交、直流工作模式及其他参数设置,可双击电压表图标,会弹出电压表特性对话框,如图 2.4.60 所示。电压表预置的内阻很高,在 $10\text{ M}\Omega$ 以上。然而,在低电阻电路中使用极高内阻的电压表,仿真时可能会产生错误。电压表特性对话框具有多种选项可供设置,包括 Label(标识)、Display(显示)、Value(数值)、Fault(故障设置)、Pin(引脚)、Variant(变量)、User fields(用户字段)内容的设置,设置方法可参阅 2.5 节中关于元器件中标签、编号、数值、模型参数的设置方法等内容。

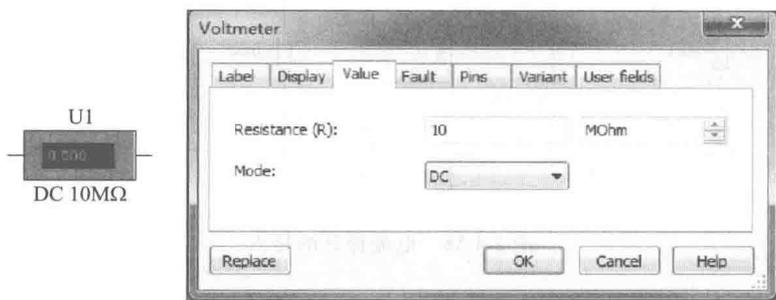


图 2.4.60 电压表的图标和特性对话框

23. 电流表

电流表存放在指示元器件库中,在使用中数量没有限制,图标和特性对话框如图 2.4.61 所示。单击旋转按钮,可以改变其引线的方向。电流表用来测量电路回路中的电流。测量时,将电流表串联接在被测电路回路中。电流表的交、直流工作模式及其他参数设置,可双击电流表图标,会弹出电流表特性对话框,如图 2.4.61 所示。电流表特性对话框具有多种选项可供设置,包括 Label(标识)、Display(显示)、Value(数值)、Fault(故障设置)、Pin(引脚)、Variant(变量)、User fields(用户字段)内容的设置,设置方法可参阅 2.5 节中关于元器件中标签、编号、数值、模型参数的设置方法等内容。

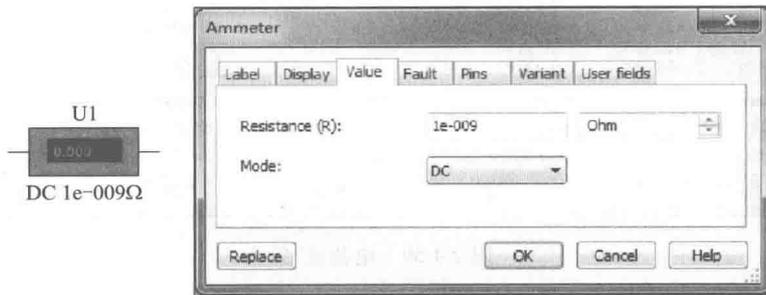


图 2.4.61 电流表的图标和特性对话框

2.5 Multisim 12 的基本操作

利用 Multisim 软件创建一个电路原理图包括建立电路文件、设置电路界面、选取并放置元器件、连接线路、编辑处理及保存文件等步骤。

一、建立电路文件

启动 Multisim 程序,则在 Multisim 基本界面上会自动打开一个空白的电路文件。在 Multisim 程序正常运行时,只需点击系统工具栏中 New 按钮,同样会出现一个空白的电路文件,系统自动命名为 Design1,可以在保存此电路文件时重新命名。

二、设置电路界面

Multisim 的基本界面好比一张制图纸,所以 Multisim 又形象地把基本界面上的原理图编辑区称为 Workspace。在进行某个实际电路实验之前,通常要定义一下制图纸张的大小、边界、电路的名称、电路的实验者及实验时间、电路中元器件的符号标准、连线的粗细、编辑区的背景及电路元件的颜色等。在 Multisim 中,可以通过 Options 菜单中的 Sheet Properties(工作界面设置)对话框中的若干个选项来实现。具体操作如下。

1. 选取 Options 中的 Sheet Properties,打开 Sheet Properties 对话框,如图 2.5.1 所示。

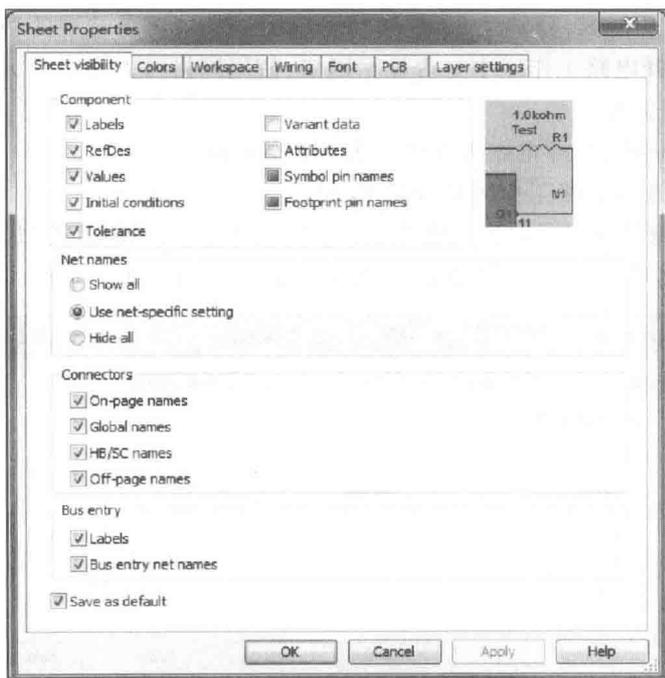


图 2.5.1 Sheet Properties 对话框

Preferences 对话框中共有 7 个选项,每个选项为单独的一个页面,包含若干个功能选项。用户通过对这 7 个选项的不同功能项的设置就可以定义一个电路界面。

2. 选择 Workspace 选项,完成对电路图纸的设置,如图 2.5.2 所示。

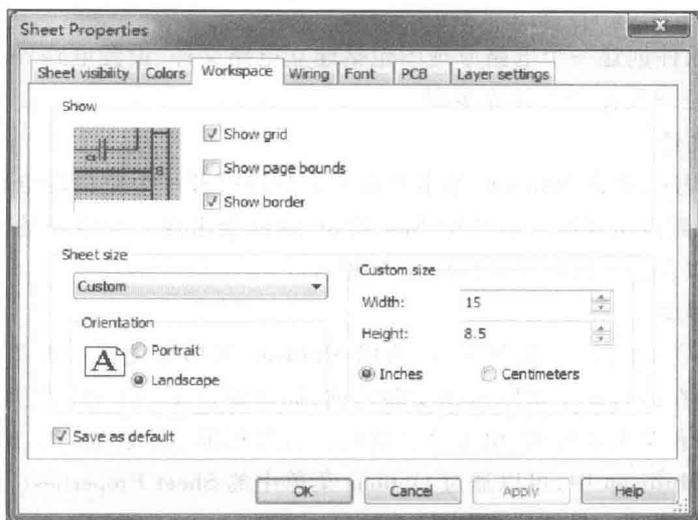


图 2.5.2 Workspace 选项

此页面上有两个功能区,分别是 Show 和 Sheet size。功能区各选项的功能如下。

- Show grid: 选择电路工作区中是否显示网格,使用网格可方便电路元器件之间的连接,使创建出的电路图整齐美观。
 - Show page bounds: 选择电路工作区是否页面分隔线。
 - Show border: 选择电路工作区是否显示边界。
 - Sheet size 区域的功能是设置图纸大小,与 Word 中的页面设置类似。
3. 选择 Wiring 选项,完成对导线和总线宽度的设置,如图 2.5.3 所示。

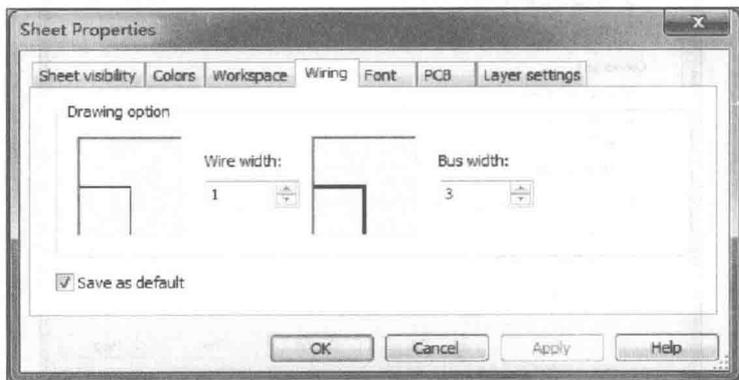


图 2.5.3 Wiring 选项

4. Sheet visibility 选项,完成对电路各种参数的设置,如图 2.5.4 所示。

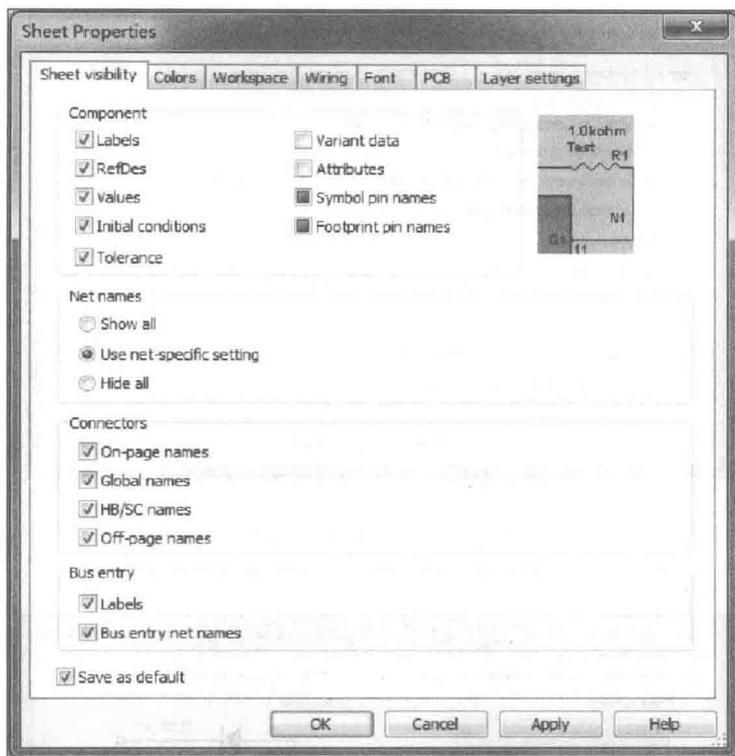


图 2.5.4 Sheet visibility 选项

- Labels:显示元器件的表示。
- RefDes:显示元器件的编号。
- Values:显示元器件数值。
- Initial conditions:选择初始化条件。

5. 选取 Options 中的 Global Preferences, 打开 Global Preferences 对话框, 然后点击 Components, 打开 Components 页, 如图 2.5.5 所示。

Components 页是对界面上元件箱出现的形式、元件箱内元件的符号标准及从元件箱中选用元件的方式的设置, 此页面有三个功能区, 分别为 Place component mode、Symbol standard 和 View。

选中 Symbol standard 区中的 DIN 项。Multisim 中提供了两套电气元器件符号标准, ANSI 是美国标准, DIN 是欧洲标准。DIN 与我国现行的标准相似, 所以选择 DIN。

在设计过程中, 为了更好地按自己的风格来进行有关设计, 需要调整一下相关窗口结构和背景设计, 可利用 Sheet Properties 菜单中的 Colors 选项进行相关设置, 如图 2.5.6 所示, 具体调试请

参照对话框中的英文说明。

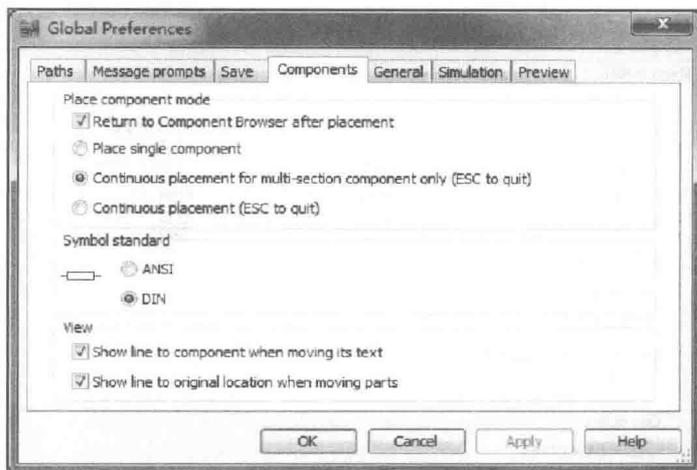


图 2.5.5 Components 选项

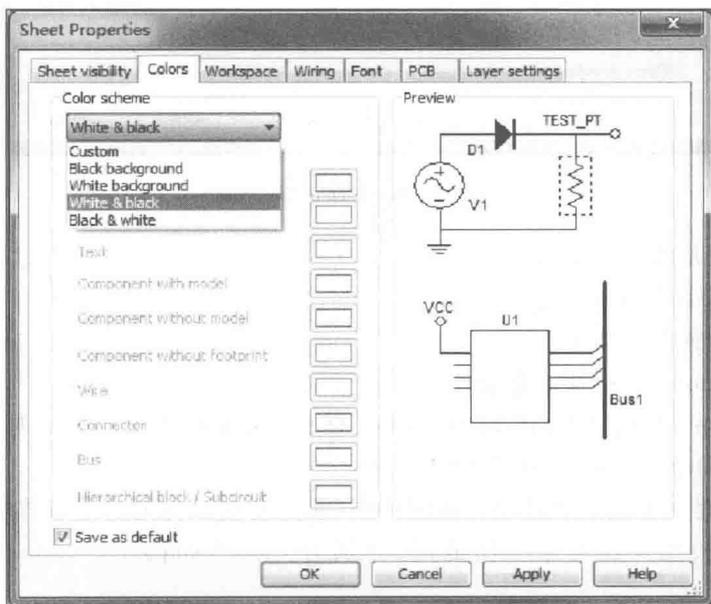


图 2.5.6 Colors 选项

经过以上简单的设置后,原理图编辑区如图 2.5.7 所示。

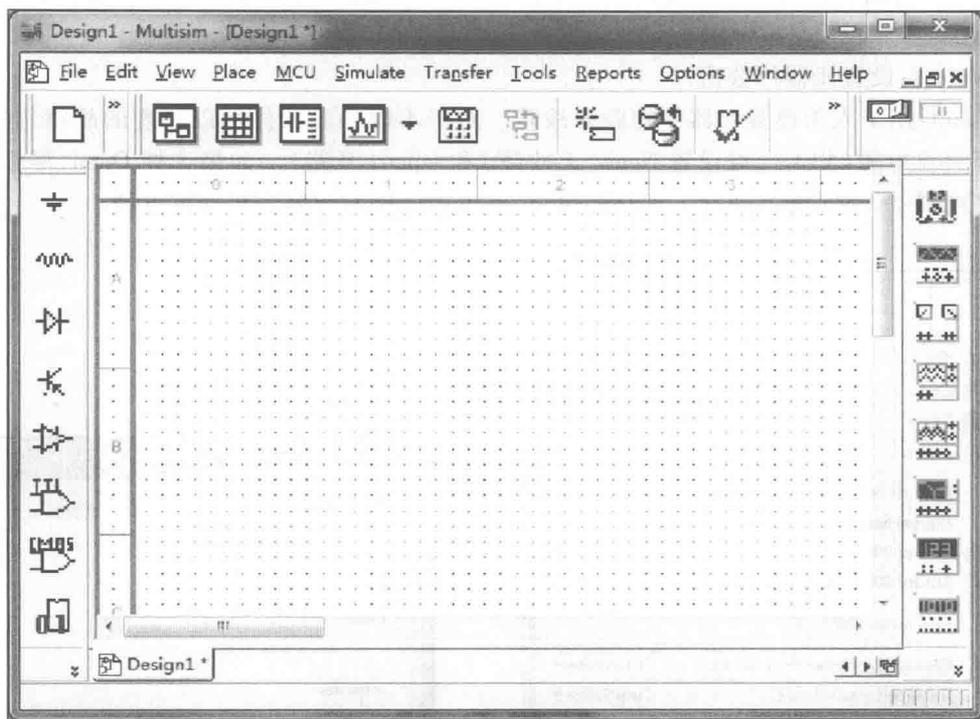


图 2.5.7 用户设置的界面

三、选取与放置元器件

1. 元器件的选取

选用元器件时,首先在元器件库中单击包含该元器件的图标,打开该元器件库。然后从选中的元器件窗口中,单击该元器件,单击“OK”按钮,用鼠标将该元器件放置在电路工作区的合适位置。

2. 元器件的复制、移动、旋转等操作

在连接电路时,要对元器件进行移动、旋转、复制、删除等操作,这就需要先选中该元器件。要选中某个元器件,单击该元器件即可。被选中的元器件四周会出现蓝色虚线方框。对选中的元件可以通过鼠标右键的选项,对其进行相关操作,如图 2.5.8 所示。

3. 元器件标签、编号、数值、模型参数的设置

在选中元器件后,双击元器件或者选择主菜单命令 Edit→Properties(元器件特性),会弹出元器件特性对话框,如图 2.5.9 所示。该对话框有多种选项可供设置,包括 Label(标签)、Display(显示)、Value(数值)、Fault(故障)、Pins(引脚端)、Variant(变量)等内容。

- Label:用于设置元器件的 RefDes(编号)和 Label(标识);RefDes(编号)由系统自动分配,必要时可以修改,但必须保证编号的唯一性。

• Display:用于设置 Label 和 RefDes 显示方式;该对话框的设置与 Option 菜单中的元器件的设置有关。

• Values:设置元器件数值。

• Fault:用于人为设置元器件的隐含故障。对于不同的元器件可以设置的故障选项不同。例如对于直流电源(VCC),可设置 None(无故障)和 Open(开路)。如果选择 Open,尽管该电源仍连接在电路中,但实际上隐含了开路的故障,这可为电路的故障分析提供方便。

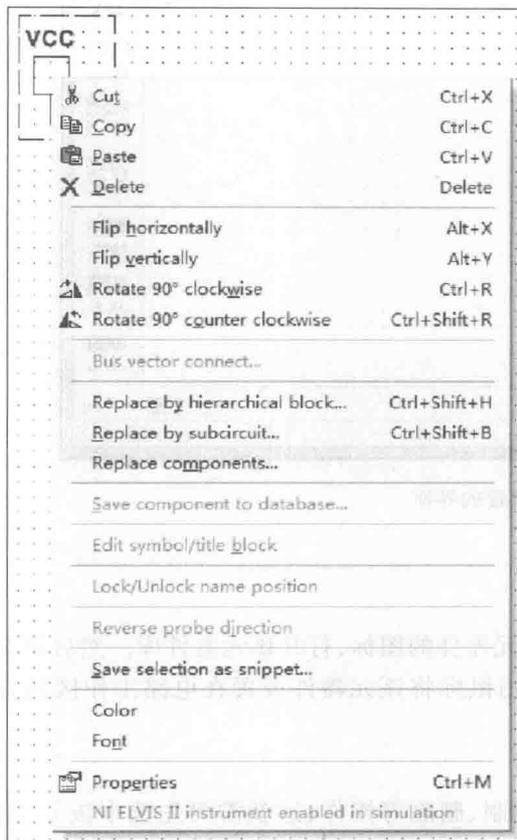


图 2.5.8 对元器件进行操作的菜单



图 2.5.9 元器件特性对话框

放置完电路中的全部元器件后,就会在 Multisim 界面上的 In Use List 栏内列出了电路所使用的所有元器件,使用它可以检查所调用的元器件是否正确。

四、线路连接及编辑处理电路中的元器件

1. 线路连接

放置完所有元器件,并对元器件标签、编号、数值、模型参数等设置后需要对其进行线路连

接。Multisim 的线路连接非常简单,将鼠标移到元件引脚处,鼠标指针就会变成小黑点,单击鼠标左键,即可拉出一条虚线;如要从某点转弯,则先点击转弯处,固定该点,然后再移动鼠标,将鼠标移到要连接的另一元件引脚处单击左键,则完成一根连线的连接,重复以上过程,画完所有连线(必须是端点连线,不能有重合的线段)。

2. 显示并修改电路的节点号

电路元件连接后,系统会自动分配给各个节点一个序号。通常这些节点序号并不出现在电路上,可启动 Options 菜单中的 Sheet Properties,打开 Sheet visibility 对话框,然后选中 Net names 中的 Show all,如图 2.5.10 所示;或者在原理图编辑区的空白处点击鼠标右键,出现一个下拉菜单,点击 Properties,也会弹出 Sheet Properties 对话框,然后进行如上的操作,点击 OK,电路的各个节点上就会显示出系统自动分配的节点号。



图 2.5.10 Net names 选项

出现在电路各节点的序号不一定是我们习惯的表示,为了便于仿真分析,可以对节点号进行修改。可将鼠标箭头对准准备修改编号的连线并双击左键,弹出 Net Properties 对话框,如图 2.5.11 所示,在 Preferred net name 中输入 vo,点击 OK,即可以将电路中的 1 号节点改为 vo。如果没有必要,这一步可以不做。

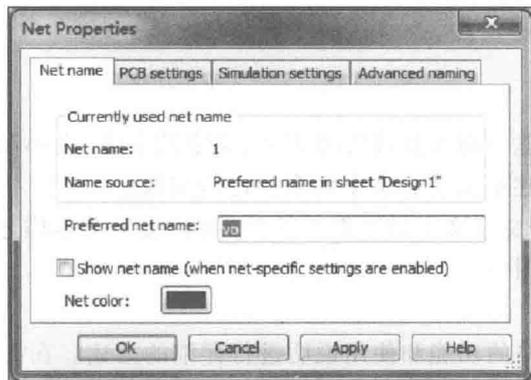


图 2.5.11 节点标号对话框

3. 改变元件和连线的颜色

在复杂的电路中,为了方便电路的连接和测试,可以将连线设置为不同颜色,可用鼠标指

向该导线,点击鼠标右键可出现菜单,选择 Segment Color,从弹出的调色板中选择合适的颜色即可。

4. 删除元件或连线

如果想删除不需要的元件或连线,可用鼠标选中该元件或连线(元件符号四周会出现虚线框或连线上出现多个小黑方块),然后点击鼠标右键可出现菜单,选择 Delete 即可。万一错删,可启动 Edit 菜单中的 Undo 命令进行恢复。另外,当删除一个元件时,与该元件连接的连线也将一并消失,但删除连线不会影响到元件。

5. “连接点”的使用

“连接点”是一个小圆点,单击 Place Junction 可以放置“连接点”。一个“连接点”最多可以连接来自 4 个方向的导线。可以直接将“连接点”插入连线中。

五、添加文本

电路图建立后,有时要为电路添加各种文本。例如,放置文字、放置电路图的标题栏以及电路描述窗等。下面简单介绍各种文本的添加方法。

1. 添加文字文本

为了便于对电路的理解,常常给局部电路添加适当的注释。允许在电路图中放置英文或中文,基本步骤如下。

(1) 单击 Place 菜单中的 Text 选项,然后单击所要放置文字文本的位置,出现如图 2.5.12 所示的文字文本描述框。

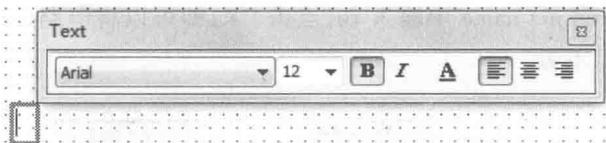


图 2.5.12 文字文本描述框

(2) 在文字文本描述框中输入要放置的文字(文字的字体、大小和颜色的设置如同 Word 文档中的操作),文字文本描述框会随着文字的多少进行缩放。

(3) 输入完毕后,单击文字文本描述框以外的界面,文字文本描述框即会消失,输入文本描述框的文字就显示在电路图中。

2. 添加电路描述框

利用电路描述框对电路的功能和使用说明进行详细的描述。在需要查看时打开,否则关闭,不会占用电路窗口有限的空间。

单击 Tool 菜单中的 Description Box Editor 选项,打开电路描述窗口,如图 2.5.13 所示,在其中可输入说明文字,还可插入图片、声音和视频。在原理图编辑区中单击 View 菜单中的 Description Box 选项,可查看电路描述窗口的内容,但不可修改。



图 2.5.13 电路描述窗口

3. 添加注释

利用注释描述框输入文本可以对电路的功能、使用进行简要说明。添加注释描述框的方法是:在需要注释的元器件旁,单击 Place 中 Comment 选项,弹出  图标,双击该图标,打开如图 2.5.14 所示的 Comment Properties 对话框,在 Comment text 区中输入文本。注释文本的字体选项可以在 Comment Properties 对话框的 Font 选项卡中设置,注释文本的放置位置及背景颜色、文本框的尺寸可以在 Comment Properties 对话框的 Display 选项卡中设置。在电路图中,在需要查看注释内容时需要将鼠标移至注释图标处,否则只显示注释图标。

4. 添加标题栏

在电路图纸的右下角常常放置一个标题栏,对电路的创建日期、创建人、校对人、使用人、图纸编号等信息进行说明。放置标题栏的方法是:单击 Place 菜单中的 Title Block...选项,弹出 NI Multisim 12 的 Title Block 子目录,在该文件夹中存放了 NI Multisim 12 为用户设计的 6 个标题栏文件和 4 个标题栏实例。选择一款默认标题 default.tb7,单击“打开”按钮,可弹出如图 2.5.15 所示的标题栏。

标题栏中包含的信息如下。

Title:电路图的标题。默认为电路的文件名。

Desc:对工程的简要描述。

Designed by:设计者的姓名。

Document No:文档编号。默认为 0001。

Revision:电路的修订次数。

Checked by:检查电路的人员姓名。

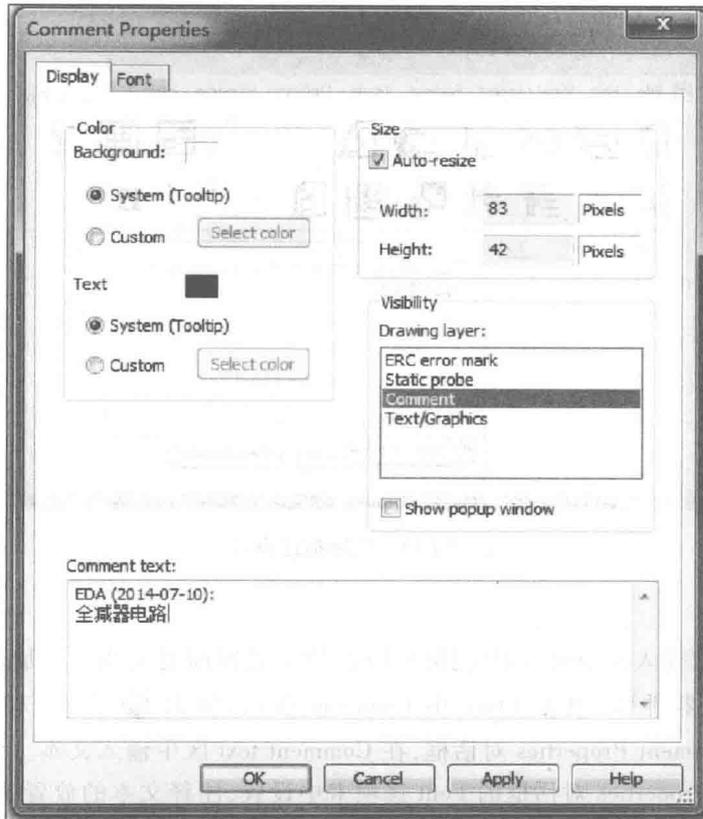


图 2.5.14 Comment Properties 对话框

National Instruments 801-111 Peter Street Toronto, ON M5V 2H1 (416) 977-5550		 NATIONAL INSTRUMENTS ELECTRONICS WORKBENCH GROUP	
Title: Design1	Desc.: Design1		
Designed by:	Document No: 0001	Revision: 1.0	
Checked by:	Date: 2014-07-10	Size: Custom	
Approved by:	Sheet 1 of 1		

图 2.5.15 NI Multisim 12 默认的标题栏

Date: 默认为电路的创建日期。

Size: 图纸的尺寸。

Approved by: 电路审批者的姓名。

Sheet 1 of 1: 当前图纸编号和图纸总数。

若要修改标题栏,只需双击,在弹出的 Title Block 对话框中进行修改即可。

六、子电路

子电路 (Subcircuit) 是由用户自己定义的一个电路 (相当于一个电路模块), 可存放在自定义元器件库供电路设计时反复调用。利用子电路可使大型的、复杂电路的设计模块化、层次化, 从而提高设计效率与设计文档的简洁性、可读性, 实现设计的重用, 缩短产品的开发周期。

为了使用子电路, 首先要创建一个子电路。下面以全减器为例详细介绍子电路的创建过程。

1. 创建子电路的电路图

按照前述的方法选取元器件、放置元器件、连线等步骤构建如图 2.5.16 所示的全减器电路原理图。

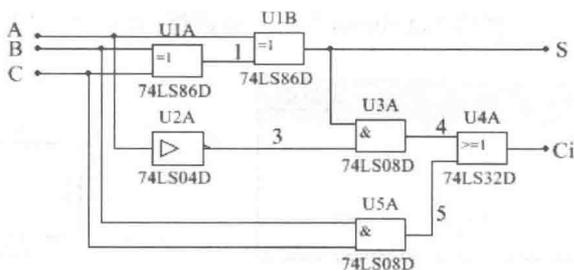


图 2.5.16 全减器电路原理图

单击 Place 菜单中的 Connectors 选项, 即可弹出添加输入/输出节点的菜单, 如图 2.5.17 所示。

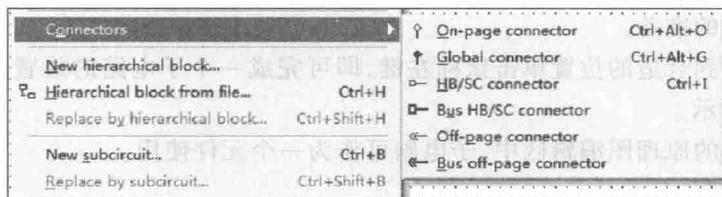


图 2.5.17 输入/输出节点菜单

在电路控制区中, 输入/输出节点可以看作只有一个引脚的元器件, 所有操作方法与元器件相同。不同的是输入/输出节点只有一个连接点。

对于本例, 需要分别给三个输入端点、两个输出端点添加输入/输出节点, 添加节点后的电路如图 2.5.18 所示。

2. 添加子电路

建立子电路的内部电路后, 下一步就是将此电路转化成一个子电路并把它放置在原理图编辑区中, 具体操作如下。

(1) 按住鼠标左键拖出一个方框, 把图 2.5.18 所示的电路图全部圈入方框内。

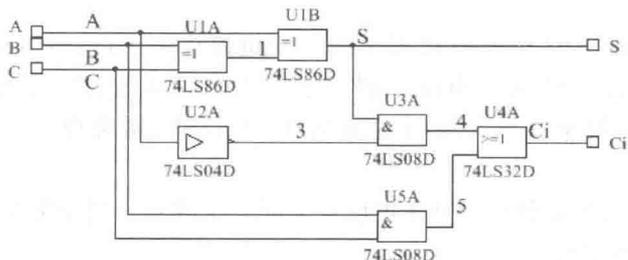


图 2.5.18 添加输入/输出节点的全减器

(2) 单击 Place 菜单中 Replace by Subcircuit 选项或单击鼠标右键,在弹出的下拉菜单中单击 Replace by Subcircuit 选项,弹出 Subcircuit Name 对话框,如图 2.5.19 所示。

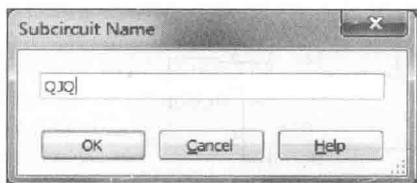


图 2.5.19 Subcircuit Name 对话框

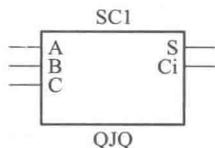


图 2.5.20 全减器子电路

(3) 在 Subcircuit Name 对话框中输入创建的子电路名称,比如 QJQ。

(4) 命名之后,单击 OK 按钮,在原理图编辑区中的鼠标箭头处出现一个尾随的虚框,表明子电路已做好放置的准备。

(5) 移动鼠标到合适的位置单击鼠标左键,即可完成一个子电路的放置。放置的全减器子电路如图 2.5.20 所示。

在含有子电路的原理图编辑区中,子电路可作为一个元件使用。

3. 子电路的编辑

在原理图编辑区中双击子电路,弹出 Hierarchical Block/Subcircuit 对话框,如图 2.5.21 所示。通过此对话框可以修改子电路的参考序列号 (Reference ID),单击 Open subsheet 按钮,可以查看和修改子电路的电路图。

在原理图编辑区中对元件的操作都适合于子电路。如添加子电路后,子电路的名称就会出现在元件列表中,如图 2.5.22 所示;选中子电路后,单击鼠标右键执行相应的菜单命令,可以对子电路进行剪切、复制、水平翻转、垂直翻转、顺时针 90° 旋转、逆时针 90° 旋转、设置颜色、字体、符号等操作。

除此之外,子电路的创建也可以采用先创建子电路符号再编辑具体电路的方法实现。单击 Place 菜单中 New Subcircuit 选项或单击鼠标右键,在 Place on Schematic 弹出的下拉菜单中单击 New Subcircuit 选项,弹出 Subcircuit Name 对话框,如图 2.5.23 所示。输入子电路的名称,点击

OK 按钮,出现如图 2.5.24 所示的子电路符号。

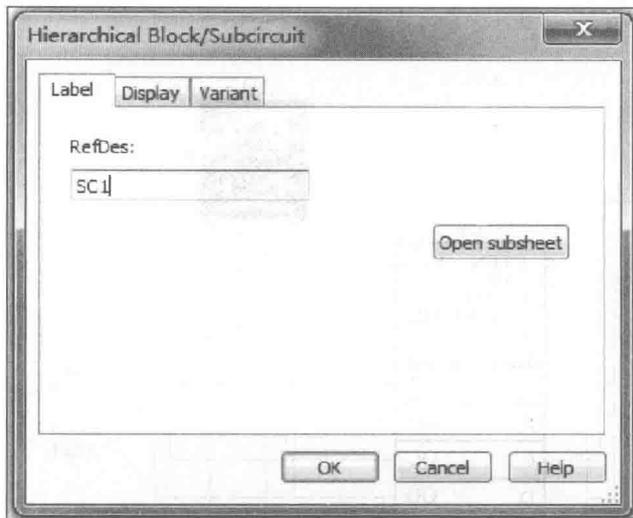


图 2.5.21 Hierarchical Block/Subcircuit 对话框



图 2.5.22 元件列表中子电路

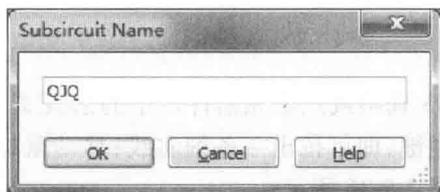


图 2.5.23 Subcircuit Name 对话框

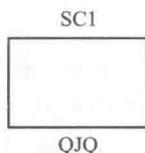


图 2.5.24 子电路符号

双击子电路符号,弹出如图 2.5.21 所示的 Hierarchical Block/Subcircuit 对话框,单击 Open subsheet 按钮,弹出子电路编辑窗口,创建子电路的电路图即选取电路元器件、连线并添加输入/输出节点,返回主电路窗口时,在主电路窗口会显示带 I/O 引脚的子电路模块。

七、总线

总线(Bus)就是用来连接一组引脚和另一组引脚的一组连线。在建立电路图时,经常会遇到一组性能相同导线的连接,如数据总线、地址总线等,当这些连接增多或距离加长时,就会使人难以分辨。如果采用总线,总线两端分别用单线连接,构成单线—总线—单线的连接方式,就会使建立的电路图简单明了。

为了说明放置和使用总线的基本方法,首先要创建一个电路。下面以 5 秒倒计时电路为例详细介绍总线的使用方法。

1. 放置总线

按照前述的方法选取元器件、放置元器件、连线等步骤构建如图 2.5.25 所示的 5 秒倒计时电路原理图。

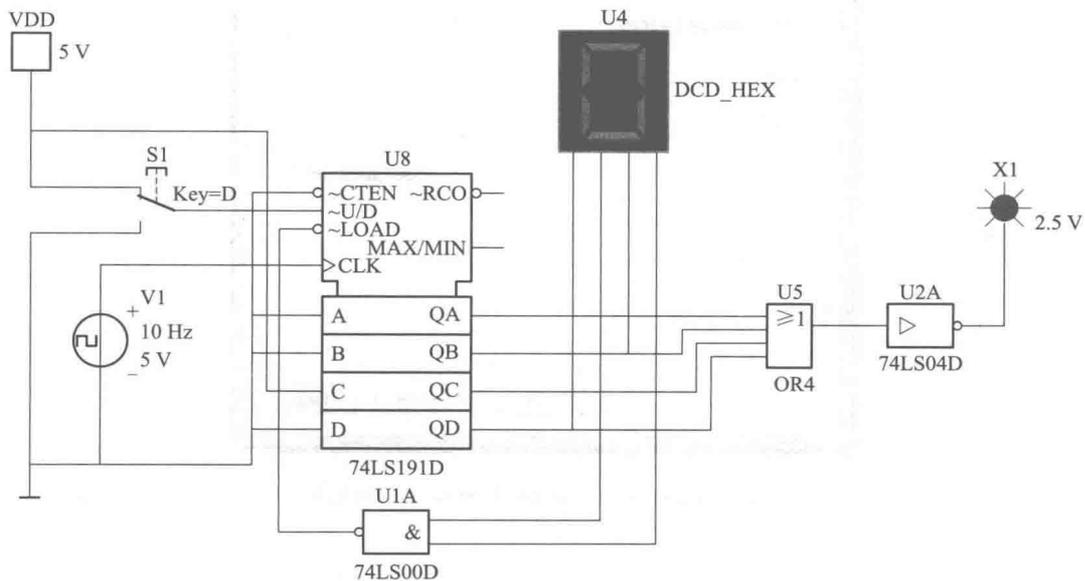


图 2.5.25 5 秒倒计时电路原理图

单击 Place 菜单中的 BUS 选项或点击元器件栏中的总线  图标,在原理图编辑区鼠标指针就会变成小黑点,单击鼠标左键,即可拉出一条细实线;移动鼠标到合适位置,双击鼠标左键,就会完成一条总线的放置,如图 2.5.26 所示。

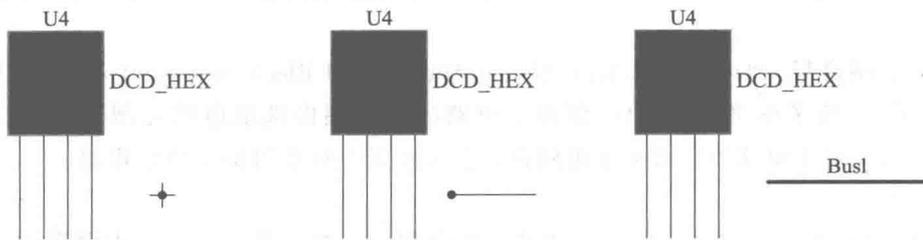


图 2.5.26 放置总线过程

2. 总线与电路的连接

将 Bus1 作为数据总线连入电路,可简化集成计数器 74LS191D 的数据输出端 QA~QD 与数码管 U4、与非门 U1A、或门 U5 之间的线路连接。具体连接方法如下。

(1) 数码管与总线的连接。

首先,断开数码管 U4 与集成计数器 74LS191D 之间的连接;然后从数码管 U4 的引脚处连接一条连线到总线 **Bus1**,接近总线时,会出现一个+45°或-45°的斜线,单击鼠标,弹出 Bus Entry Connection 对话框,将 Bus line 修改为 QD,即表明 U4 的这一引脚要与 74LS191D 的数据输出端 QD 相连;修改完毕后,单击 OK 按钮,完成引脚到总线的连接,如图 2.5.27 所示。数码管 U4 其他引脚与总线的连接与上述步骤类似。

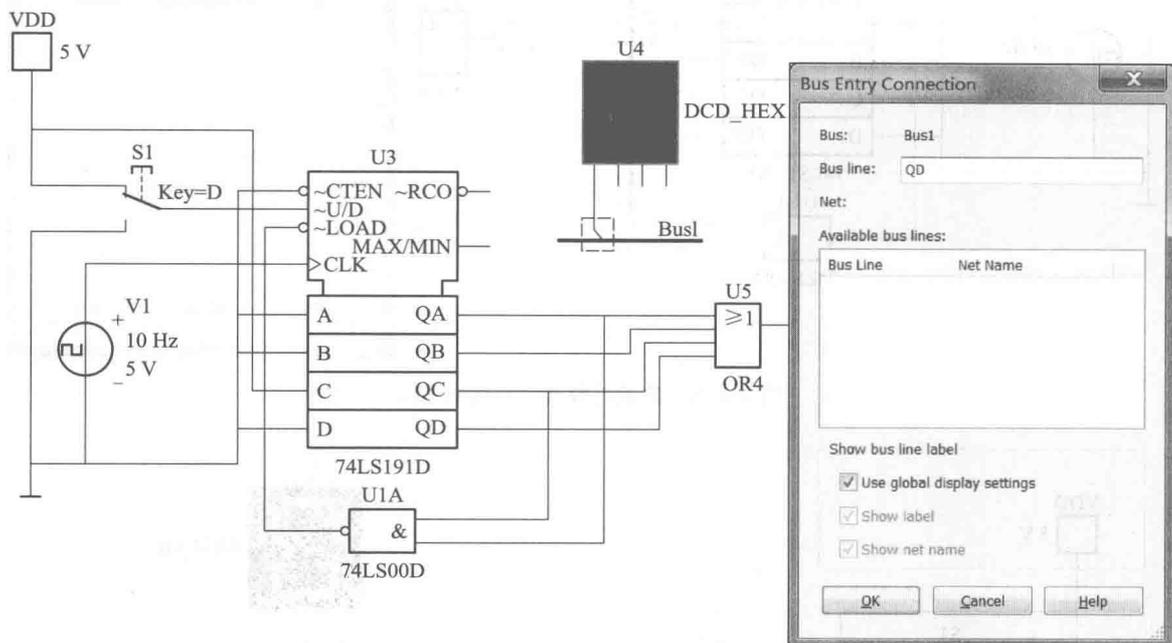


图 2.5.27 数码管与总线的连接

(2) 集成计数器 74LS191D 与总线的连接。

首先,将鼠标移至总线 **Bus1** 的左侧端点,单击鼠标左键并向下移动鼠标到合适的位置后,双击鼠标左键,完成 **Bus1** 的延长;其次,断开集成计数器 74LS191D 与门电路之间的连接;然后将 74LS191D 的 QA 端与总线连接,在弹出 Bus Entry Connection 对话框中的可选总线连接线 (Available bus lines) 选项中选择 QA,单击 OK 按钮,完成引脚到总线的连接,如图 2.5.28 所示。

(3) 按照上述方法完成电路其他部分与总线的连接,并按照子电路的创建方法(详见 2.5 节中关于子电路的介绍)将电路中的四输入或门 U5 和非门 U2A 用子电路“四输入或非门”代替,得到含有总线结构和子电路的 5 秒倒计时电路原理图,如图 2.5.29 所示。

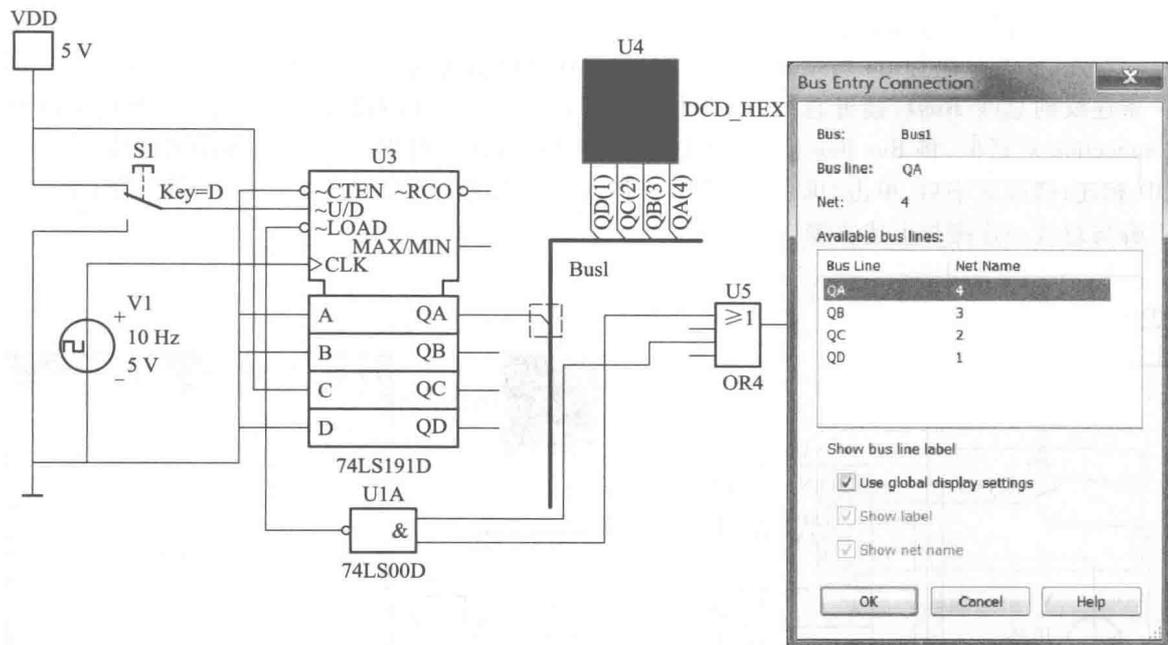


图 2.5.28 集成计数器与总线的连接

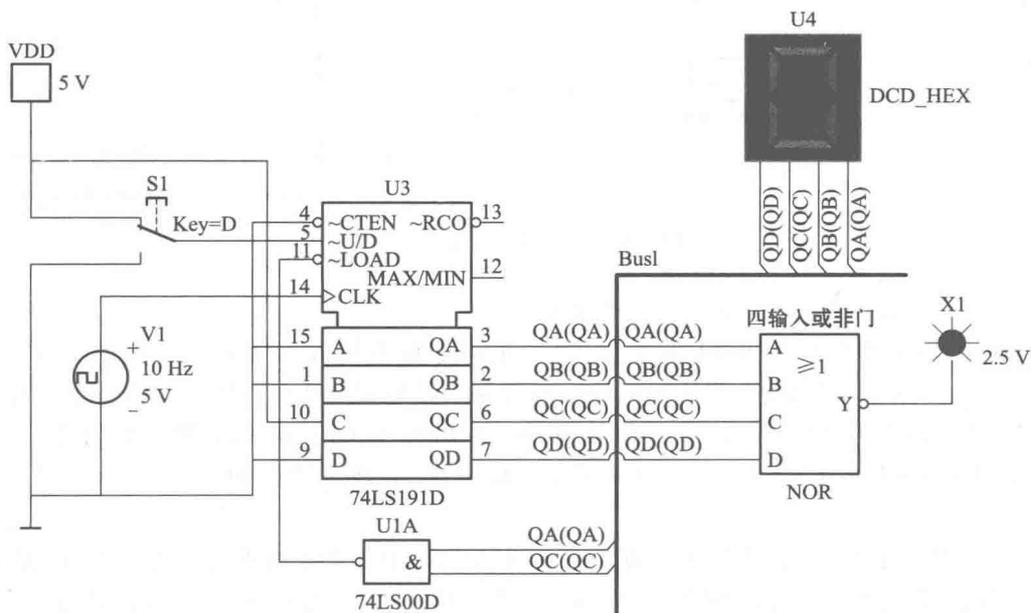


图 2.5.29 含有总线结构和子电路的 5 秒倒计时电路原理图

八、保存文件

1. 选择菜单项 File|Save 或点击工具栏中的存储快捷键  ,弹出保存文件对话框。
2. 选定保存文件的路径。
3. 在文件名编辑框中输入文件名,如:erjiguan。
4. 单击“保存”按钮。

刚才绘制的电路图就保存在选定的路径中,以 erjiguan.ms12 为文件名的文件。

第三章

电子技术综合实验 Multisim 仿真部分

题目一 移位寄存器型彩灯控制器

彩灯控制器可以自动控制多路彩灯按不同的节拍循环显示各种灯光变换花型,它广泛用于大型灯会、舞台灯光控制、体育馆灯光控制以及节假日灯光装饰中。实现彩灯控制的方法很多,如用可编程器件、单片机、纯硬件电路等。当彩灯数较少且花型变换较为简单时,可采用移位寄存器进行控制。

一、设计任务与要求

(1) 要求采用移位寄存器设计一个彩灯控制器。

(2) 要求三种变换花型。

以某种节拍按一定规律改变彩灯的输入电平值,控制彩灯的亮与灭,按预定规律显示一定的花型。

花型 1:8 路彩灯从左至右顺次渐亮,全亮后逆序渐灭;循环两次。

花型 2:8 路彩灯从中间到两边对称地逐次渐亮,全亮后仍由中间到两边逐次渐灭;循环两次。

花型 3:8 路彩灯分两半,从左至右顺次渐亮,全亮后则灭,循环三次。

花型真值表如表 3.1.1 所示。

表 3.1.1 花型真值表

节拍顺序	编码 Q_A Q_B Q_C Q_D Q_E Q_F Q_G Q_H							
	花型 1	花型 2	花型 3					
0	00000000	00000000	00000000					
1	10000000	00011000	10001000					
2	11000000	00111100	11001100					
3	11100000	01111110	11101110					
4	11110000	11111111	11111111					
5	11111000	11100111						
6	11111100	11000011						
7	11111110	10000001						
8	11111111							
9	11111110							
10	11111100							
11	11111000							
12	11110000							
13	11100000							
14	11000000							
15	10000000							

当然三种花型也可以与上述相反。

花型 1:8 路彩灯从右至左顺次渐亮,全亮后逆序渐灭;循环两次。

花型 2:8 路彩灯从两边到中间对称地逐次渐亮,全亮后仍由两边到中间逐次渐灭;循环两次。

花型 3:8 路彩灯分两半,从右至左顺次渐亮,全亮后则灭,循环三次。

(3) 三种花型先以慢节拍显示一遍,再以快节拍显示一遍,构成一次大循环,然后重复。当然也可以相反。

二、总体方案设计

1. 原理框图

彩灯控制器以某种节拍按一定规律改变彩灯的输入电平值,控制彩灯的亮与灭,按预定规律显示一定的花型。因此彩灯控制器需要一个能够按一定规律输出不同高低电平编码信号的编码发生器,同时还需要编码发生器所要求的时序信号和控制信号。由于实际应用场合所用彩灯可能是功率较大的白炽灯或其他,因此还需要一定的驱动电路,实验中因采用发光二极管或指示灯,故可省略驱动电路。彩灯控制器的原理框图如图 3.1.1 所示。

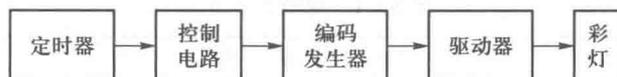


图 3.1.1 彩灯控制器原理框图

2. 设计思路

(1) 节拍控制电路

三种花型循环一次需要 $16 \times 2 + 8 \times 2 + 5 \times 3 = 63$ 个节拍,再加上清零一拍,共 64 拍。三种花型若先以慢节拍显示一遍,再以快节奏显示一遍,这样构成一次大循环共需 128 节拍。快慢节拍的时钟电路框图如图 3.1.2 所示。

CP_L (慢)的 1—32 节拍显示花型 1,其中 1—8 节拍移位寄存器右移 8 位,9—16 节拍移位寄存器左移 8 位,17—32 节拍重复 1—16 节拍内容。 CP_L (慢)的 33—48 节拍显示花型 2,其中 33—40 节拍移位寄存器 1 左移,移位寄存器 2 右移,41—48 节拍重复 33—40 节拍内容。 CP_L (慢)



图 3.1.2 快慢节拍的时钟电路框图

的 49—63 节拍显示花型 3,其中 49—53 节拍 2 个移位寄存器右移,54—58、59—63 节拍重复 49—53 节拍内容。 CP_L (慢)的 64 节拍总清零。 CP_H (快)的 64 节拍与慢节拍显示相同。因此,节拍控制电路需要产生 8、16、32、64、128 拍的节拍控制信号。节拍控制信号电路框图如图 3.1.3 所示,节拍控制信号时序图如图 3.1.4 所示。



图 3.1.3 节拍控制信号电路框图

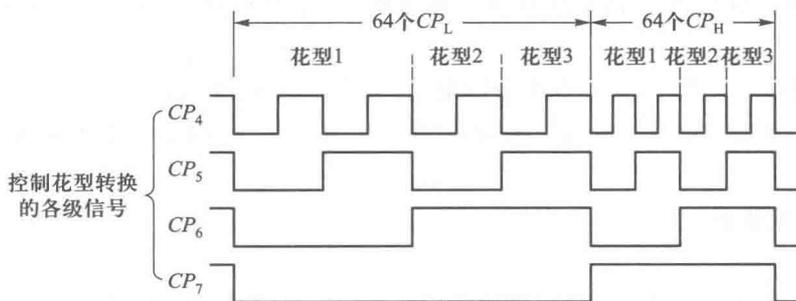


图 3.1.4 节拍控制信号时序图

$CP_7 = 0$ 时,移位脉冲 CP 为慢节拍 CP_L ; $CP_7 = 1$ 时, CP 为快节奏 CP_H 。所以能够选择快慢节拍的移位脉冲 CP 的表达式为

$$CP = \overline{CP_7} \cdot CP_L + CP_7 \cdot CP_H$$

CP_7 的周期为 128 节拍。

(2) 编码发生器

通用移位寄存器 74LS194 的功能表如表 3.1.2 所示。

表 3.1.2 74LS194 的功能表

\overline{Cr}	S_1	S_0	CP	功能
0	×	×	×	异步清零
1	0	0	↑	保持
1	0	1	↑	右移
1	1	0	↑	左移
1	1	1	↑	预置

要实现上述花型显示,关键是移位寄存器的功能控制端 S_0 、 S_1 和左移 SL 、右移 SR 的信号电平如何组织。

(3) 清零电路

清零控制电路如图 3.1.5 所示。清零电路的功能如下。

① 系统总清零,用启动开关 S 控制。

② 花型 3 每 5 拍清零一次, $Cr = \overline{Q_E} \cdot Q_H$ 。

③ 三种花型运行一遍即第 64 拍时总清零一次,可用对周期为 64 拍的 CP_6 微分得到的负脉冲实现。

三、设计报告要求(设计报告格式见附录 F)

(1) 目的与要求:根据任务书进行填写。

(2) 正文:简单叙述设计过程(包括原理、方案);画出完整的电路原理图,并简述各部分的功能。

(3) 总结与结论:写出设计和调试过程中出现的问题及解决方法;简述心得体会。

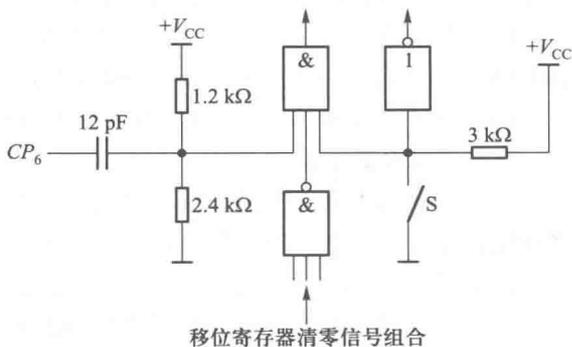


图 3.1.5 清零控制电路

题目二 智力竞赛抢答器

在进行智力竞赛抢答题比赛时,各参赛者考虑好后都想抢先回答,如果没有合适的设备,主持人难以分清抢答者的先后。为了使比赛能顺利进行,需要有一个能判断抢答者先后的设备,称为智力竞赛抢答器。常见的智力竞赛抢答器如图 3.2.1 所示。

一、设计任务与要求

(1) 设计制作一个可容纳六组参赛队的智力竞赛抢答器,每组设置一个抢答按钮供抢答者使用。设置一个“系统复位”或“抢答准备命令”按钮和一个“抢答开始命令”按钮供主持人使用。

(2) 电路具有第一抢答信号的鉴别和锁存功能。在主持人将系统复位并发出“抢答开始命令”后,若参赛者按下抢答按钮,就显示最先抢答者的组号,指示抢答有效,并以声音警示。若系统复位但未发“抢答开始命令”参赛者就按下抢答按钮,也显示抢答者的组号,但指示抢答无效,并以声音警示。要求确定第一个输入的抢答信号,并保持该信号不变,同时使后输入的信号无效。

(3) 在发出“抢答开始命令”后开始计时,经过规定的抢答时间后若没有人抢答,就发出“抢答时间到”信号,以声光警示,并锁定输入电路使各路抢答信号无法再输入。

(4) 设置计分电路,开始时每组预置为 100 分或其他,答对一次加 10 分,答错减 10 分。

二、总体方案设计

(1) 本题的根本任务是准确判断出第一抢答者的信号并将其锁存。实现这一功能可用触发器或锁存器等。在得到第一抢答信号之后应立即将电路的输入封锁,使其他组的抢答信号无效。同时还必须注意,第一抢答信号应该在主持人发出“抢答开始命令”后才有效,否则无效。

(2) 当电路形成第一抢答信号之后,用编码、译码及数码显示电路显示出抢答者的组号,也可用发光二极管直接指示出组号。还可用鉴别出的第一抢答信号控制音频振荡器工作,给以警示。

(3) 计分电路可采用 2 位七段数码管显示,由于每次都是加或减 10 分,故个位总保持为 0,只要十位和百位作加减计数即可,可采用两级十进制加减计数器完成。

三、设计报告要求(设计报告格式见附录 F)

(1) 目的与要求:根据任务书进行填写。

(2) 正文:简单叙述设计过程(包括原理、方案);画出完整的电路原理图,并简述各部分的功能。

(3) 总结与结论:写出设计和调试过程中出现的问题及解决方法;简述心得体会。

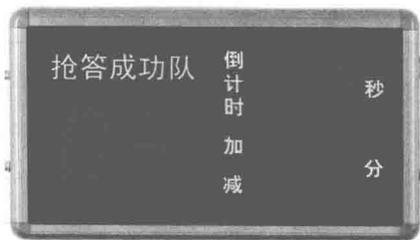


图 3.2.1 智力竞赛抢答器

题目三 电子拔河游戏机

一、设计任务与要求

电子拔河游戏机是一种能容纳甲乙双方参赛或甲乙双方加裁判的三人游戏电路。由一排 LED 发光二极管表示拔河的“电子绳”。由甲、乙二人通过按钮开关使发光的 LED 管向自己一方的终点移动,当亮点移到任何一方的终点时,则该方获胜,连续比赛多局以定胜负。

(1) 由裁判下达比赛开始命令后,甲乙双方才能输入信号,否则,由于电路具有自锁功能,使输入信号无效。

(2) “电子绳”至少由 17 个 LED 管构成,裁判下达“准备开始”命令后,位于“电子绳”中点的 LED 管发亮。裁判下达“比赛开始”命令后,甲、乙二人通过按钮开关使发光的 LED 管向自己一方的终点移动,并阻止其向对方延伸。当点亮的 LED 管到达某一方的终点时,该方获胜。此时通过自锁功能锁定电路,使输入不再有效,必须由裁判再次发出“比赛开始”命令时方能开始下一次比赛。

(3) 某方赢一次,由计分电路自动给该方加分一次,通过多次比赛以定胜负。

二、总体方案设计

电子拔河游戏机原理框图如图 3.3.1 所示。

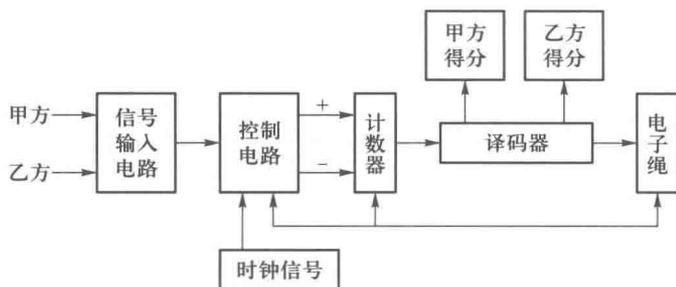


图 3.3.1 电子拔河游戏机原理框图

信号输入电路与控制电路的设计,要求甲乙的输入互相控制,即在有效时间里若甲先输入则甲有效,乙先输入则乙有效。由两端输出,无效端保持不变,有效端输出一脉冲进入计数器的加减脉冲输入端,其进入方向则由参赛双方输入的按键信号决定。

计数电路由加减计数器构成,加减两端分别接控制门的两个输出端,若当甲有效时,就进行加计数,乙有效时则减计数。

译码器的输入接计数器的输出,输出接电子绳,根据加减计数情况决定电子绳的亮灯向甲方移动还是向乙方移动。

计分器的设计,当甲方灯亮的时候,给甲加一分,乙方灯亮的时候,给乙加一分。并能对多次

比赛计分,最后计算总成绩。当甲方或乙方最后一个灯亮的时候,电路被锁定,使输入不再有效,当裁判清零以后方能进行下一次比赛。

三、设计报告要求(设计报告格式见附录 F)

(1) 目的与要求:根据任务书进行填写。

(2) 正文:简单叙述设计过程(包括原理、方案);画出完整的电路原理图,并简述各部分的功能。

(3) 总结与结论:写出设计和调试过程中出现的问题及解决方法;简述心得体会。

题目四 交通信号灯控制器

在经济飞速发展的今天,城乡交通自动指挥越来越显得重要,为了确保行人和各种车辆运行安全,保证正常的交通秩序,应对十字路口的红绿灯进行自动控制。实现红绿灯的自动指挥是城市管理自动化的重要课题。

一、设计任务与要求

交通信号灯示意图如图 3.4.1 所示。

在一个具有主干道、支干道的十字路口,设计一个交通灯自动控制装置。红灯(R)亮表示该条道路禁止通行,黄灯(Y)亮作为过渡,以使行驶中的车辆有时间停到禁止线以外,绿灯(G)亮表示该条道路允许通行。具体要求如下。

- (1) 主干道经常通行。
- (2) 支干道有车才通行。
- (3) 主干道、支干道均有车时,两者交替通行,并要求主干道每次至少放行 30 s,支干道每次最多放行 20 s;采用传感器分别监测主干道、支干道是否有车,有车时向主控制器发出信号,实验中用逻辑开关代替。

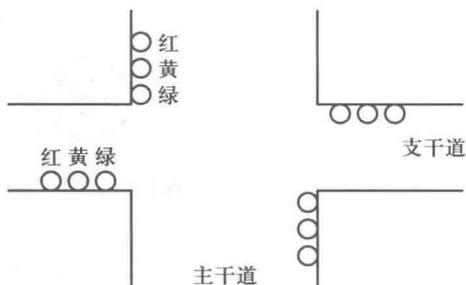


图 3.4.1 交通信号灯示意图

- (4) 每次绿灯变红灯时,要求黄灯先亮 5 s,此时原红灯不变。
- (5) 设计 5 s、20 s、30 s 计时的译码显示电路,每秒改变一次显示数字,可以采用正计时,也可倒计时。
- (6) 当任意一条路上出现特殊情况,如消防车、救护车或其他需要优先放行的车辆时,各方向上均为红灯,计时停止。当特殊运行状态结束后,控制器恢复原来状态,继续正常运行。用逻辑开关模拟有无特殊情况。

二、总体方案设计

(1) 交通灯自动控制系统原理框图如图 3.4.2 所示。

采用传感器分别监测主、支干道是否有车,有车时向主控制器发出信号,实验中用逻辑开关代替。

时钟信号发生器产生稳定的“秒”脉冲($f=1\text{ Hz}$)信号,确保整个电路装置同步工作和实现定时控制。

计时器按设计要求记录“秒”脉冲个数完成计时任务,并向主控制器发出相应的定时信

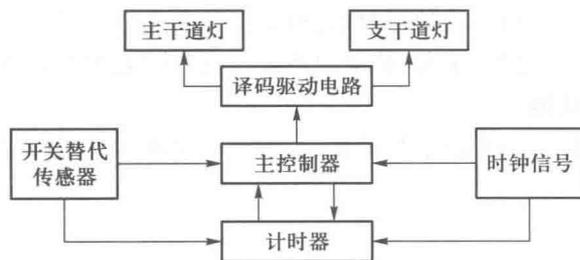


图 3.4.2 交通灯自动控制系统原理框图

号以控制各干道通车时间和黄灯亮时间。

主控制器根据传感器和计时器送来的信号,保持或改变电路的状态,以实现各干道车辆运行状态的控制。

按照主控制器所处的状态进行译码,再驱动相应的信号灯,指挥各干道的行人和车辆。

(2) 交通灯控制的状态设计。

交通灯控制的状态转换图如图 3.4.3 所示,状态转换表如表 3.4.1 所示。

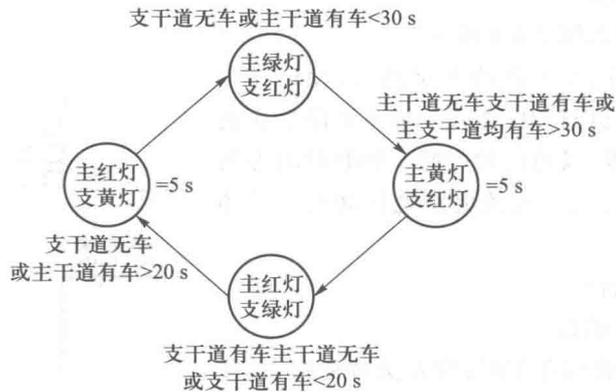


图 3.4.3 交通灯控制的状态转换图

表 3.4.1 交通灯状态表

状态	主干道	支干道
状态 1(S_0)	绿	红
状态 2(S_1)	黄	红
状态 3(S_2)	红	绿
状态 4(S_3)	红	黄

三、设计报告要求(设计报告格式见附录 F)

(1) 目的与要求:根据任务书进行填写。

(2) 正文:简单叙述设计过程(包括原理、方案);画出完整的电路原理图,并简述各部分的功能。

(3) 总结与结论:写出设计和调试过程中出现的问题及解决方法;简述心得体会。

题目五 电子秒表

电子秒表是一种较先进的电子计时器,目前国产的电子秒表一般都是利用石英振荡器的振荡频率作为时间基准,采用6位液晶数字显示时间。它不仅能显示分、秒,还能显示时、日、月及星期,并且有 $(1/100)\text{s}$ 的功能。一般的电子秒表连续累计时间为 $59\text{ min }59.99\text{ s}$,可读到 $(1/100)\text{ s}$,平均日差 $\pm 0.5\text{ s}$ 。

电子秒表配有三个按钮,分别为秒表按钮、功能变换按钮和调整按钮,基本显示的计时状态为“时”、“分”、“秒”,如图3.5.1所示。

一、设计任务与要求

- (1) 利用555定时器制作一个频率为500 Hz时钟发生装置。
- (2) 通过分频电路,输出周期为0.01 s的计数脉冲。
- (3) 利用计数器接受分频电路输出的计数脉冲,并通过数码显示器显示出来,完成0.01~99.99 s的秒表计时。

(4) 使用基本RS触发器及其他外围电路制作电子秒表的控制开关,实现“开始计数”、“停止并保持计数”和“清零并准备重新开始计数”的功能,在秒表计数期间应使“开始计数”和“清零并准备重新开始计数”无效。

二、总体方案设计

1. 整体设计布局。

整体设计布局如图3.5.2所示。

2. 模块总体方案设计图

(1) 555时钟发生模块

利用555定时器实现的多谐振荡电路能够完成时钟信号发生器的功能,通过调节电路中的可变电阻使多谐振荡器的输出信号频率为500 Hz。

(2) 分频电路模块

利用74LS290将输入为500 Hz频率的时钟脉冲进行分频变为100 Hz的信号输出,即周期为0.01 s。

(3) 输出及显示模块

四个74LS290分别连接成十进制计数器,一个输出百分之一秒,一个输出十分之一秒,一个输出整秒个位,一个输出整秒十位,并通过四个数码显示器显示0.01~99.99 s的秒表数值。

(4) 控制电路

利用基本RS触发器生成控制电路:S有效,则Q端输出高电平,控制时钟信号输出到分频电

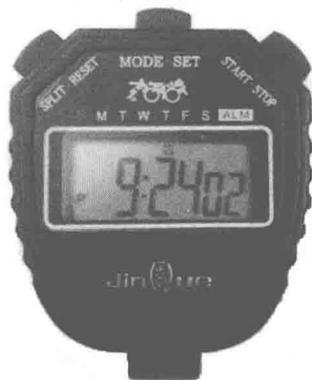


图 3.5.1 电子秒表

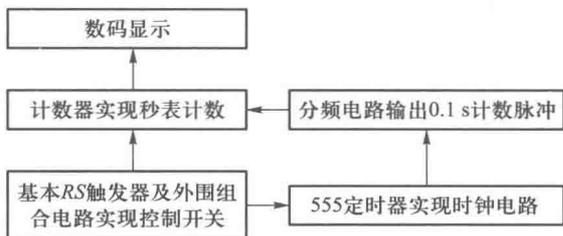


图 3.5.2 整体设计框图

路,实现“开始计数”;同时 Q 非端输出的低电平使“清零并准备重新开始计数”无效; R 有效,则时钟信号被屏蔽,电子秒表保持当前数值不变,同时允许“清零并准备重新开始计数”信号输入。注意,设计时要避免基本 RS 触发器出现不定态。

三、设计报告要求(设计报告格式见附录 F)

- (1) 目的与要求:根据任务书进行填写。
- (2) 正文:简单叙述设计过程(包括原理、方案);画出完整的电路原理图,并简述各部分的功能。
- (3) 总结与结论:写出设计和调试过程中出现的问题及解决方法;简述心得体会。

题目六 电子密码锁

一、设计任务与要求

锁是人们生活中的常用物品。本题要求用电子器件设计制作一个密码锁,使之在输入正确的代码时,输出开锁信号以推动执行机构动作,并用红灯亮、绿灯灭表示关锁,而绿灯亮、红灯灭表示开锁。

(1) 在锁的控制电路中储存一个可修改的 8421BDC 码作为密码,当输入代码和锁的预置密码相等时,进入开锁状态使锁打开。

(2) 从第一次密码输入之后的 5 s 内若未将锁打开,则电路进入自锁状态,使之无法再打开,并由扬声器发出持续 10 s 的报警信号。

二、总体方案设计

1. 电子密码锁的原理框图。

电子密码锁的原理框图如图 3.6.1 所示。

2. 设计思路

(1) 该题的主要任务是产生一个开锁信号,而开锁信号的形成条件是输入代码和已设置的密码相同。实现这种功能的电路构思有多种。比如:用 2 片 8 位数据锁存器或 2 片 4 位寄存器,一片存入开锁的代码,另一片存入预置密码,通过比较的方法判断,若二者相等,则形成开锁信号。

(2) 在产生开锁信号后,要求输出声、光信号。其中音响的产生可以由开锁信号去触发一个音响电路;其中光信号可以用开锁信号点亮 LED 指示灯。

(3) 用按钮开关的第一个动作信号触发一个 5 s 的定时器,若在 5 s 内未将锁打开,则电路进入自锁状态,使之无法再打开,并由扬声器发出持续 10 s 的报警信号。

三、设计报告要求(设计报告格式见附录 F)

(1) 目的与要求:根据任务书进行填写。

(2) 正文:简单叙述设计过程(包括原理、方案);画出完整的电路原理图,并简述各部分的功能。

(3) 总结与结论:写出设计和调试过程中出现的问题及解决方法;简述心得体会。

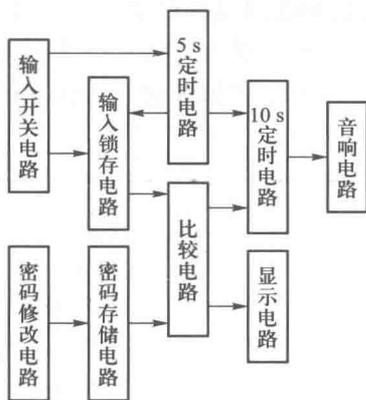


图 3.6.1 电子密码锁的原理框图

题目七 数字电子钟

电子钟是一种高精度的计时工具,它采用了集成电路和石英晶体技术,因此走时精度高,稳定性能好,使用方便,且不需要经常调校。电子钟根据显示方式不同,分为指针式电子钟和数字式电子钟。指针式电子钟采用机械传动带动指针显示;而数字式电子钟则是采用译码电路驱动数码显示器件,以数字形式显示。这些译码显示器件,利用集成技术可以做得非常小巧,也可以另加一定的驱动电路,推动霓虹灯或白炽灯显示系统,制作成大型电子钟表。因此,数字式电子钟用途非常广泛。

一、设计任务与要求

(1) 设计一个能直接显示时、分、秒,并具有校时功能的数字电子钟。小时采用二十四进制。

(2) 设计 24 小时整点报时控制电路,要求每整点发出一声音响报时;要求只在 6:00—22:00 之间每整点报时一次,23:00—5:00 之间整点不报时。

(3) 设计任意几点几分均可响铃的闹钟控制电路。响铃 1 分钟,可提前终止。

(4) 根据规定的作息时间表,如表 3.7.1 所示,设计自动响铃控制电路。

表 3.7.1 作息时间表

作息时间表	
起床	6:50
上午上班	8:00
午饭	11:45
下午上班	13:30
下班	17:30

二、总体方案设计

数字式电子钟的基本功能是能够实现时、分、秒的正确计时,计时单位为 1 s。因此,一个简单的数字式电子钟,首先必须有计时显示电路和秒脉冲产生电路。其次,当刚接通电源或时钟走时出现误差时,需要进行时间校准,否则就不能正确表示当前时间。因此,数字式电子钟应有校时电路。另外,若要求数字电子钟能够自动整点报时(参考电路见图 3.7.2,此电路是当时钟到达 59 分 50 秒时开始报时,每 2 秒报一次。)或按要求时间闹铃,还应有整点报时和闹铃控制电路。若还需要其他功能,还要有一些相应的控制电路。综上所述,数字式电子钟应由三大部分组成:计时显示电路、秒脉冲产生电路和控制电路。在软件设计调试中脉冲产生电路可以省略,用时钟信号源代替。其结构框图如图 3.7.1 所示。

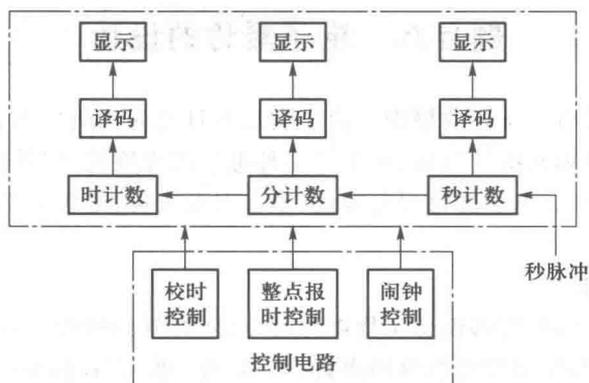


图 3.7.1 数字式电子钟的结构框图

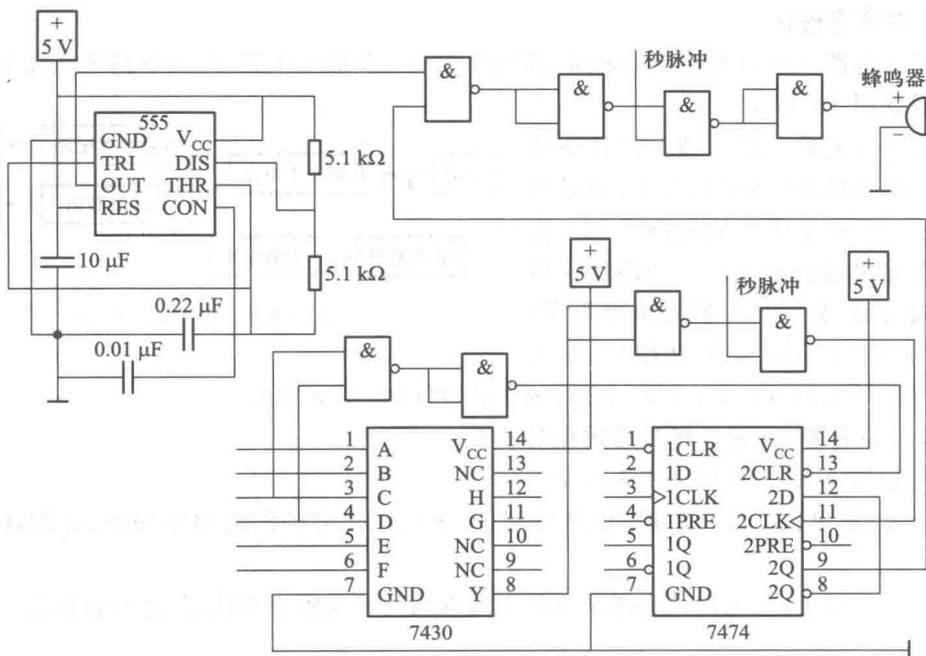


图 3.7.2 整点报时电路(仅供参考)

三、设计报告要求(设计报告格式见附录 F)

- (1) 目的与要求: 根据任务书进行填写。
- (2) 正文: 简单叙述设计过程(包括原理、方案); 画出完整的电路原理图, 并简述各部分的功能。
- (3) 总结与结论: 写出设计和调试过程中出现的问题及解决方法; 简述心得体会。

题目八 电子爆竹的设计

目前,各大中城市纷纷禁止燃放鞭炮。为了满足节日及喜庆场所人们(特别是小孩)燃放鞭炮的强烈愿望,增加节日和喜庆的气氛,诞生了一种电子闪光鞭炮,它外形酷似鞭炮,能发出逼真响亮的燃放鞭炮声与光彩。电子爆竹无危险性,不会引发火灾,不污染空气,可重复多次使用,是传统鞭炮的理想替代品。

一、设计任务与要求

设计一个电子爆竹电路,能够模仿实物爆竹的点燃、燃放的过程。具体要求如下。

- (1) 电路工作时能够发出鞭炮燃放的声音,如啪-啪-啪,声音频率控制在 100 Hz 左右。
- (2) 电路工作时能够出现鞭炮燃放时的闪光效果,闪光频率为 200 Hz。
- (3) 电路有清零、重启功能。

二、总体方案设计

电子爆竹电路一般由光控触发电路、音频产生及音频放大电路、闪光电路等几部分组成。整体设计布局如图 3.8.1 所示。

接通电源开关后,光敏电阻(设计中可以用滑线变阻器代替)的阻值变小,通过放大电路放大,驱动音频产生电路工作。音频产生电路输出的音效电信号一路经音频放大电路放大后,驱动扬声器发出鞭炮声;另一路作为计数器的计数脉冲信号,使其输出端依次循环输出高电平,发光二极管依次轮流导通,轮流发光。

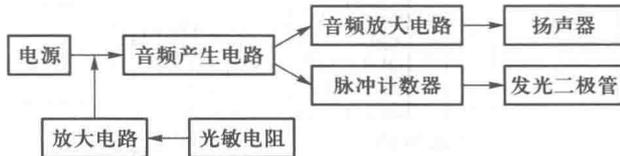


图 3.8.1 整体设计框图

三、设计报告要求(设计报告格式见附录 F)

- (1) 目的与要求:根据任务书进行填写。
- (2) 正文:简单叙述设计过程(包括原理、方案);画出完整的电路原理图,并简述各部分的功能。
- (3) 总结与结论:写出设计和调试过程中出现的问题及解决方法;简述心得体会。

题目九 数字频率计

数字频率计是用于测量信号(方波、正弦波或其他脉冲信号)的频率,并用十进制数字显示的数字测量仪器,它具有精度高、测量迅速、读数方便等优点。

一、设计任务与要求

- (1) 利用 555 定时器设计秒时间基准发生器。
- (2) 使用组合逻辑电路对输入信号进行放大整形。
- (3) 通过 4 个十进制计数/锁存/七段译码/驱动器 CD40110 计数并显示 0~9 999 Hz 信号的频率,并能通过增加 CD40110 对更高频率的信号进行计数显示。

二、总体方案设计

总体框图如图 3.9.1 所示。

1. 秒时间基准发生器

利用 555 定时器组成多谐振荡器,根据振荡频率公式 $f = 1.443 / (R_A + 2R_B) C$,调节相应元器件的参数值使输出频率为 1 Hz,注意利用可变电阻来精调振荡频率。

输出信号经电容和非门/与非门进行整形后作为控制电路的输入信号。

2. 控制电路

控制电路由 CMOS 集成电路 CD4017 构成,其引脚分布如图 3.9.2(a) 所示,其中 C_r 为清零端,当高电

平或者正脉冲出现在该引脚时,输出端 $Q_0 \sim Q_9$ 全部为 0。两个时钟输入端 CP 和 \overline{EN} ,当 \overline{EN} 为低电平时,上升沿计数,信号由 CP 端输入,当 CP 为高电平时,下降沿计数,信号由 \overline{EN} 输入。CD4017 有 10 个译码输出端,每个输出端的状态与输入计数器的时钟脉冲的个数相对应。例如输入 4 个时钟脉冲,如果计数器从 0 开始计数,则此时译码输出端 Q_4 应为高电平,其余输出端均为低电平。每输入 10 个时钟脉冲, Q_{10} 输出一个进位脉冲。

控制电路如图 3.9.2(b) 所示,阀门控制信号控制与非门的一端,与非门的另一端为被测信号输入端。计数显示控制信号控制 CD40110 的清零端,掌握计数显示与清零重新计数的交替进行。

3. 输入信号的放大整形

根据所学知识自行设计输入信号的放大整形电路,放大信号时要防止信号幅值过大,损坏器件。

4. 计数显示电路

CD40110 引脚分布如图 3.9.3 所示。

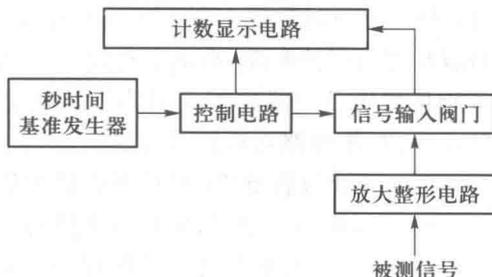


图 3.9.1 数字频率计的总体框图

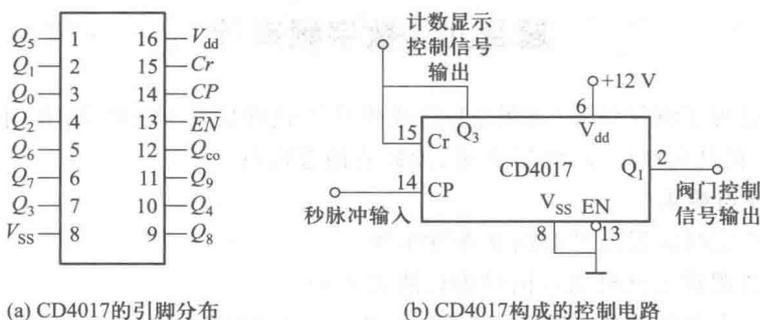


图 3.9.2 CD4017 的引脚分布和控制电路

R 为清零端, $R=1$ 时, 计数器复零; CP 为时钟输入端 (CP_u : 加法计数时钟; CP_d : 减法计数时钟); Q_{co} 输出进位脉冲, Q_{bo} 输出借位脉冲; $\overline{TE}=0$ 时计数器工作, $\overline{TE}=1$ 时, 计数器处于禁止状态, 输出保持; LE 为锁存控制端, 高电平有效。 V_{dd} 接 +12 V 电源, V_{ss} 接地。 $a \sim g$ 分别接数码显示器的相应引脚。计数显示电路由 4 个 CD40110 级联而成, 具体电路自行设计完成。

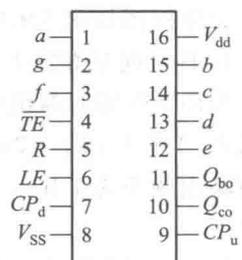


图 3.9.3 CD40110 的引脚分布

三、设计报告要求(设计报告格式见附录 F)

- (1) 目的与要求: 根据任务书进行填写。
- (2) 正文: 简单叙述设计过程(包括原理、方案); 画出完整的电路原理图, 并简述各部分的功能。
- (3) 总结与结论: 写出设计和调试过程中出现的问题及解决方法; 简述心得体会。

题目十 用运算放大器组成万用电表的设计与调试

在测量电路的各种参数时,电表的接入应不影响被测电路的原工作状态,这就要求电压表应具有无穷大的输入电阻,电流表的内阻应为零。但实际上,万用电表表头的可动线圈总有一定的电阻,例如 $100\ \mu\text{A}$ 的表头,其内阻约为 $1\ \text{k}\Omega$,用它进行测量时将影响被测量,引起误差。此外,交流电表中的整流二极管的压降和非线性特性也会产生误差。如果在万用电表中使用运算放大器,就能大大降低这些误差,提高测量精度。在欧姆表中采用运算放大器,不仅能得到线性刻度,还能实现自动调零。

一、设计任务与要求

- (1) 直流电压表 满量程 $+6\ \text{V}$ 。
- (2) 直流电流表 满量程 $10\ \text{mA}$ 。
- (3) 交流电压表 满量程 $6\ \text{V}$, $50\ \text{Hz}\sim 1\ \text{kHz}$ 。
- (4) 交流电流表 满量程 $10\ \text{mA}$ 。
- (5) 欧姆表 满量程分别为 $1\ \text{k}\Omega$, $10\ \text{k}\Omega$, $100\ \text{k}\Omega$ 。

二、总体方案设计

1. 直流电压表

为了减小表头参数对测量精度的影响,将表头置于运算放大器的反馈回路中,这时,流经表头的电流与表头的参数无关,只要切换 $R_1\sim R_3$ 就可进行量程的切换。

高精度直流电压表原理图如图 3.10.1 所示。

图 3.10.1 所示电路适用于测量电路与运算放大器共地的有关电路。此外,当被测电压较高时,在运放的输入端应设置衰减器。

2. 直流电流表

图 3.10.2 是浮地直流电流表的原理图。在电流测量中,浮地电流的测量是普遍存在的,例如:若被测电流无接地点,就属于这种情况。为此,应把运算放大器的电源也对地浮动,按此种方式构成的电流表就可像常规电流表那样,串联在任何电流通路中测量电流。

表头电流 I 与被测电流 I_1 间关系为

$$I = \left(1 + \frac{R_1}{R_2} \right) I_1$$

可见,改变电阻比 $\frac{R_1}{R_2}$,可调节流过电流表的电流,以提

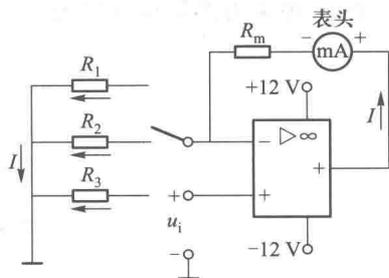


图 3.10.1 高精度直流电压表原理图

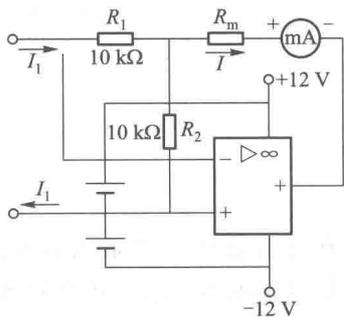


图 3.10.2 浮地直流电流表原理图

高灵敏度。如果被测电流较大时,应给电流表表头并联分流电阻。

3. 交流电压表

由运算放大器、二极管整流桥和直流毫安表组成的交流电压表如图 3.10.3 所示。被测交流电压 u_i 加到运算放大器的同相端,故有很高的输入阻抗,又因为负反馈能减少反馈回路中的非线性影响,故把二极管桥路和表头置于运算放大器的反馈回路中,以减小二极管本身非线性的影响。

电流 I 全部流过桥路,其值仅与 U_i/R_1 有关,与桥路和表头参数(如二极管的死区等非线性参数)无关。表头中电流与被测电压 u_i 的全波整流平均值成正比,若 u_i 为正弦波,则表头可按有效值来刻度。被测电压的上限频率决定于运算放大器的频带和上升速率。

4. 交流电流表

根据直流电流表电路与交流电压表电路的工作原理,自行设计交流电流表。

5. 欧姆表

图 3.10.4 为多量程欧姆表。

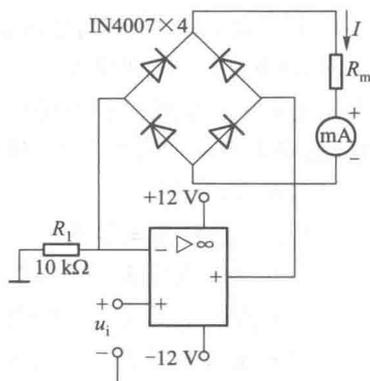


图 3.10.3 交流电压表原理图

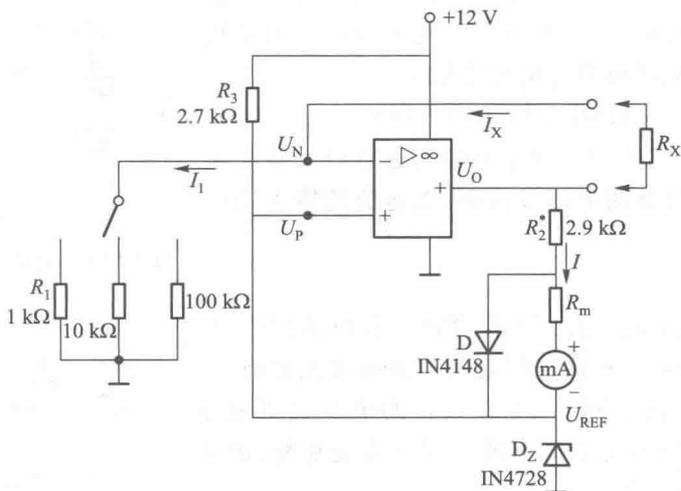


图 3.10.4 多量程欧姆表电路

在此电路中,运算放大器改由单电源供电,被测电阻 R_x 跨接在运算放大器的反馈回路中,同相端加基准电压 U_{REF} 。由电路可知

$$R_x = \frac{R_1}{U_{\text{REF}}}(U_0 - U_{\text{REF}})$$

$$I = \frac{U_0 - U_{\text{REF}}}{R_2 + R_m}$$

两式消去 $(U_0 - U_{\text{REF}})$, 可得

$$I = \frac{U_{\text{REF}} R_x}{R_1 (R_m + R_2)}$$

可见电流 I 与被测电阻成正比, 而且表头具有线性刻度, 改变 R_1 值, 可改变欧姆表的量程。这种欧姆表能自动调零, 当 $R_x = 0$ 时, 电路变成电压跟随器, $U_0 = U_{\text{REF}}$, 故表头电流为零, 从而实现了自动调零。

二极管 D 起保护电表的作用, 如果没有 D , 当 R_x 超量程时, 特别是当 $R_x \rightarrow \infty$, 运算放大器的输出电压将接近电源电压, 使表头过载。有了 D 就可使输出钳位, 防止表头过载。调整 R_2 , 可实现满量程调节。

三、设计报告要求(设计报告格式见附录 F)

- (1) 目的与要求: 根据任务书进行填写。
- (2) 正文: 简单叙述设计过程(包括原理、方案); 画出完整的电路原理图, 并简述各部分的功能。
- (3) 总结与结论: 写出设计和调试过程中出现的问题及解决方法; 简述心得体会。

题目十一 温度检测及控制电路

一、设计任务与要求

- (1) 设计由双臂电桥和差动输入集成运放组成的桥式放大电路。
- (2) 掌握滞回比较器的性能和调试方法。
- (3) 将 1、2 两部分电路根据相应特性和原理结合起来组成温度检测电路。
- (4) 学会系统测量和调试,并加入控制电路。

二、总体方案设计

1. 含有热敏电阻的桥式放大电路

(1) 测温电桥

如图 3.11.1 所示, R_1 、 R_2 、 R_3 、 R_{P1} 及 R_4 组成测温电桥, 其中 R_1 是温度传感器。其呈现出的阻值与温度成线性变化关系且具有负温度系数, 而温度系数又与流过它的工作电流有关。为了稳定 R_1 的工作电流, 达到稳定其温度系数的目的, 设置了稳压管 D_2 。 R_{P1} 可决定测温电桥的平衡。测温电桥的输出电压为 $\Delta U = U_B - U_A$ 。

(2) 差动放大电路

如图 3.11.2 所示, 由 A_1 及外围电路组成的差动放大电路, 将测温电桥输出电压 ΔU 按比例放大。其输出电压当 $R_4 = R_5$ 、 $(R_7 + R_{P2}) = R_6$ 时

$$U_{O1} = \frac{R_7 + R_{P2}}{R_4} (U_B - U_A)$$

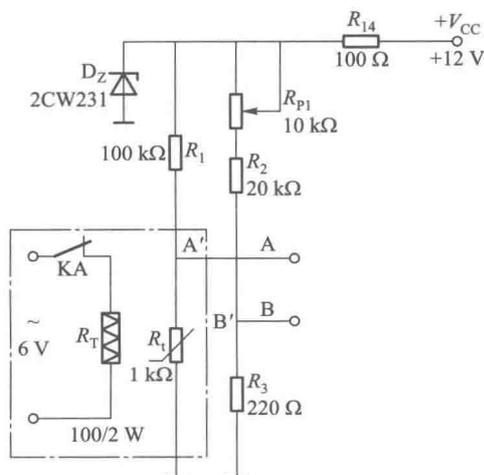


图 3.11.1 测温电桥电路

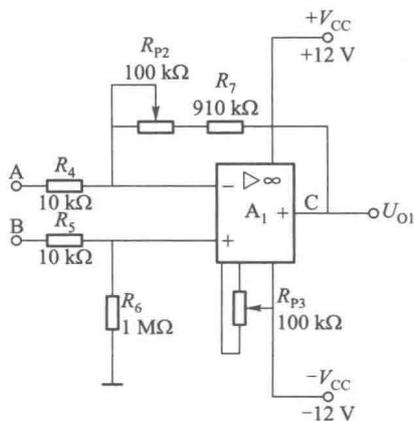


图 3.11.2 差动放大电路

可见差动放大电路的输出电压 U_{O1} 仅取决于两个输入电压之差和外部电阻的比值。 R_{P3} 用于差动放大器调零。

2. 滞回比较器

差动放大器的输出电压 U_{O1} 输入由 A_2 组成的滞回比较器。

滞回比较器的单元电路如图 3.11.3 所示, 设比较器输出高电平为 U_{OH} , 输出低电平为 U_{OL} , 参考电压 U_R 加在反相输入端。

其电压传输特性如图 3.11.4 所示, 门限宽度 $\Delta U_T = U_{TH} - U_{TL} = \frac{R_2}{R_F}(U_{OH} - U_{OL})$, 公式结果自行推导。门限宽度的大小可通过调节 R_2/R_F 的比值来调节。

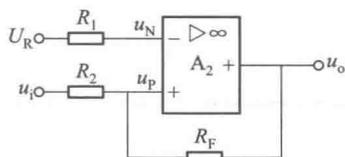


图 3.11.3 滞回比较器的单元电路

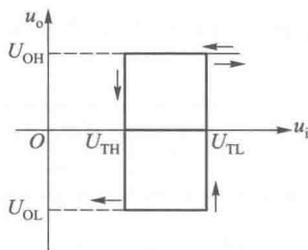


图 3.11.4 滞回比较器的电压传输特性

3. 输出报警和控制电路

利用滞回比较器输出的电压 U_{O2} 控制一个开关晶体管使报警电路中的发光二极管显示不同的状态(亮/灭), 同时控制电流继电器 KA, 进而控制加热电路的导通和截止。调节滞回比较器的上下门限电平可控制晶体管的开关时间, 从而达到设定加热温度的目的。电路如图 3.11.5 所示。

4. 系统测量和调试

(1) 差动放大器

① 运放调零。将 A、B 两端对地短路, 调节 R_{P3} 使 $U_0 = 0$ 。

② 去掉 A、B 端对地短路线。从 A、B 端分别加入不同的两个直流电平。

当电路中 $R_7 + R_{P2} = R_6$, $R_4 = R_5$ 时, 其输出电压

$$u_o = \frac{R_7 + R_{P2}}{R_4} (U_B - U_A)$$

在测试时, 要注意加入的输入电压不能太大, 以免放大器输出进入饱和区。

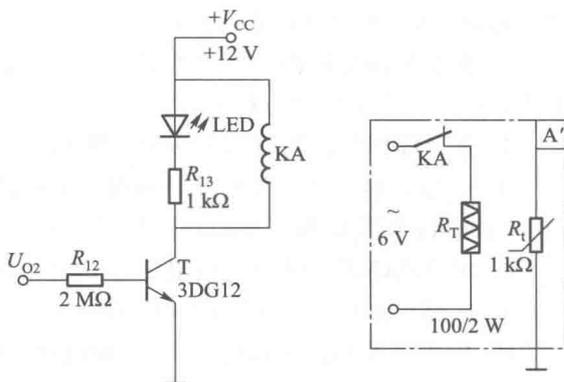


图 3.11.5 输出报警和控制电路

③ 将 B 点对地短路,把频率为 100 Hz、有效值为 10 mV 的正弦波加入 A 点。用示波器观察输出波形。在输出波形不失真的情况下,用交流毫伏表测出 u_i 和 u_o 的电压。算得此差动放大电路的电压放大倍数 A 。

(2) 桥式测温放大电路

将差动放大电路的 A、B 端与测温电桥的输出端相连,构成一个桥式测温放大电路。

① 在室温下使电桥平衡

在实验室室温条件下,调节 R_{p1} ,使差动放大器输出 $U_{o1} = 0$ (注意:前面实验中调好的 R_{p3} 不能再动)。

② 温度系数 $K(V/^\circ\text{C})$

由于测温需升温槽,为使实验简易,可虚设室温 T 及输出电压 u_{o1} ,温度系数 K 也定为一个常数,具体参数由读者自行填入表格 3.11.1 内。

表 3.11.1

温度 $T/^\circ\text{C}$	室温/ $^\circ\text{C}$				
输出电压 U_{o1}/V	0				

从表 3.11.1 中可得到 $K = \Delta U / \Delta T$ 。

③ 桥式测温放大器的温度-电压关系曲线

根据前面测温放大器的温度系数 K ,可画出测温放大器的温度-电压关系曲线,实验时要标注相关的温度和电压的值,如图 3.11.6 所示。从图中可求得在其他温度时,放大器实际应输出的电压值。也可得到在当前室温时, U_{o1} 实际对应值 U_s 。

④ 重调 R_{p1} ,使测温放大器在当前室温下输出 U_s 。即调 R_{p1} ,使 $U_{o1} = U_s$ 。

(3) 温度检测控制电路整机工作状态

① 连接各级电路。(注意:可调元件 R_{p1} 、 R_{p2} 、 R_{p3} 不能随意变动。如有变动,必须重新进行前面内容。)

② 根据所需检测报警或控制的温度 T ,从测温放大器温度-电压关系曲线中确定对应的 u_{o1} 值。

③ 调节滞回比较器参考电压 U_R ,使 $U_R = U_{o1}$ 。

④ 用加热器升温,观察温升情况,直至报警电路动作报警

(在实验电路中当 LED 发光时作为报警),记下动作时对应的温度值 t_1 和 U_{o11} 的值。

⑤ 用自然降温法使热敏电阻降温,记下电路解除时所对应的温度值 t_2 和 U_{o12} 的值。

⑥ 根据 t_1 和 t_2 值,可得到检测灵敏度 $t_0 = (t_2 - t_1)$ 。

注:实验中的加热装置可用一个 $100 \Omega/2 \text{ W}$ 的电阻 R_T 模拟,将此电阻靠近 R_1 即可。

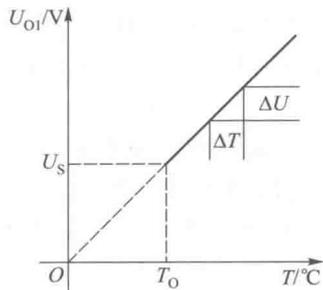


图 3.11.6 温度-电压关系曲线

三、设计报告要求(设计报告格式见附录 F)

- (1) 目的与要求:根据任务书进行填写。
- (2) 正文:简单叙述设计过程(包括原理、方案);画出完整的电路原理图,并简述各部分的功能。
- (3) 总结与结论:写出设计和调试过程中出现的问题及解决方法;简述心得体会。

题目十二 随机存取存储器 2114A 及其应用

随机存取存储器(RAM),又称读写存储器,它能存储数据、指令、中间结果等信息。在该存储器中,任何一个存储单元都能以随机次序迅速地存入(写入)信息或取出(读出)信息。随机存取存储器具有记忆功能,但停电(断电)后,所存信息(数据)会消失,不利于数据的长期保存,所以多用于中间过程暂存信息。

一、设计任务与要求

(1) 通过学习 2114A 的工作原理和功能以及引脚分布,结合基本 RS 触发器与八缓冲器/线驱动器/线接收器 74LS244 设计出静态随机存取器。

(2) 以静态随机存取器为基础,在电路中添加 8 线-3 线优先编码器 74LS148 和 4 位二进制同步计数器 74LS163(同步清零)来实现静态顺序存取功能。

二、总体方案设计

1. RAM 的结构、工作原理

图 3.12.1 是 RAM 的基本结构图,它主要由存储单元矩阵、地址译码器和读写控制电路三部分组成。



图 3.12.1 RAM 的基本结构图

(1) 存储单元矩阵

存储单元矩阵是 RAM 的主体,一个 RAM 由若干个存储单元组成,每个存储单元可存放 1 位二进制数或 1 位二元代码。为了存取方便,通常将存储单元设计成矩阵形式,所以称为存储矩阵。存储器中的存储单元越多,存储的信息就越多,表示该存储器容量就越大。

(2) 地址译码器

为了对存储矩阵中的某个存储单元进行读出或写入信息,必须首先对每个存储单元的所在位置(地址)进行编码,然后当输入一个地址码时,就可利用地址译码器找到存储矩阵中相应的一个(或一组)存储单元,以便通过读/写控制,对选中的一个(或一组)单元进行读出或写入信息。

(3) 片选与读/写控制电路

由于集成度的限制,大容量的 RAM 往往由若干片 RAM 组成。当需要对某一个(或一组)存储单元进行读出或写入信息时,必须首先通过片选 CS,选中某一片(或几片),然后利用地址译码

器才能找到对应的具体存储单元,以便读/写控制信号对该片(或几片)RAM 的对应单元进行读出或写入信息操作。

2. 2114A 的引脚分布与功能

2114A 是一种 1024 字×4 位的静态随机存取存储器,采用 HMOS 工艺制作,它的引脚排列如图 3.12.2 所示,表 3.12.1 是引出端功能表。

其中,有 4096 个存储单元排列成 64×64 矩阵。采用两个地址译码器,行译码($A_3 \sim A_8$)输出 $X_0 \sim X_{63}$,从 64 行中选择指定的一行,列译码(A_0, A_1, A_2, A_9)输出 $Y_0 \sim Y_{15}$,再从已选定的一行中选出 4 个存储单元进行读/写操作。

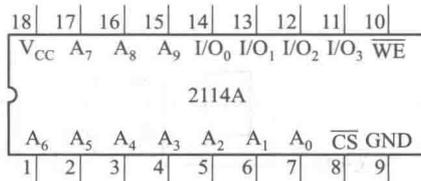


图 3.12.2 2114A 的引脚排列

$I/O_0 \sim I/O_3$ 既是数据输入端,又是数据输出端, \overline{CS} 为片选信号, \overline{WE} 是写使能,控制器件的读写操作,表 3.12.2 是器件的状态表。

表 3.12.1 2114A 引出端功能

端名	功能
$A_0 \sim A_9$	地址输入端
\overline{WE}	写选通
\overline{CS}	芯片选择
$I/O_0 \sim I/O_3$	数据输入/输出端
V_{cc}	+5 V

表 3.12.2 2114A 状态表

地址	\overline{CS}	\overline{WE}	$I/O_0 \sim I/O_3$
有效	1	×	高阻态
有效	0	1	读出数据
有效	0	0	写入数据

(1) 当器件要进行读操作时,首先输入要读出单元的地址码($A_0 \sim A_9$),并使 $\overline{WE} = 1$,给定地址存储单元内容(4 位)就经读写控制传送到三态输出缓冲器,而且只能在 $\overline{CS} = 0$ 时才能把读出数据送到引脚($I/O_0 \sim I/O_3$)上。

(2) 当器件要进行写操作时,在 $I/O_0 \sim I/O_3$ 端输入要写入的数据,在 $A_0 \sim A_9$ 端输入要写入单元的地址码,然后再使 $\overline{WE} = 0, \overline{CS} = 0$ 。必须注意,在 $\overline{CS} = 0$ 时, \overline{WE} 输入一个负脉冲,则能写入信息;同样, $\overline{WE} = 0$ 时, \overline{CS} 输入一个负脉冲,也能写入信息。因此,在地址码改变期间, \overline{WE} 或 \overline{CS} 必须至少有一个为 1,否则会引起误写入,冲掉原来的内容。为了确保数据能可靠地写入,写脉冲宽度 t_{wp} 必须大于或等于手册所规定的时间区间,当写脉冲结束时,就标志这次写操作结束。

3. 用 2114A 静态随机存取存储器实现数据的随机存取

(1) 作为读写控制开关的基本 RS 触发器

为保证 RS 触发器不出现不定态的情况,触发器的设计如图 3.12.3 所示。

(2) 八缓冲器/线驱动器/线接收器 74LS244 的引脚分布及功能

74LS244 引脚分布如图 3.12.4 所示。

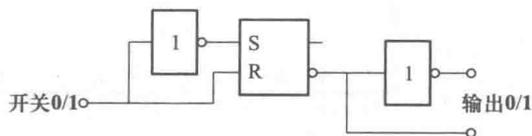


图 3.12.3 触发生器的设计

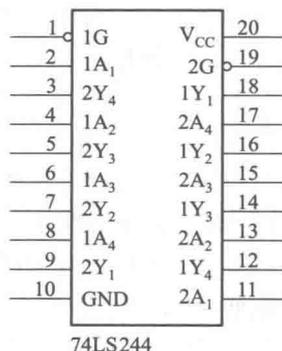


图 3.12.4 74LS244 的引脚分布

$1A_1 \sim 1A_4 / 2A_1 \sim 2A_4$: 输入端。

$1\bar{G} / 2\bar{G}$: 三态允许端(低电平有效)。

$1Y_1 \sim 1Y_4 / 2Y_1 \sim 2Y_4$: 输出端。

功能如表 3.12.3 所示。

表 3.12.3 74LS244 的功能

输入		输出
\bar{EN}	A	Y
0	0	0
0	1	1
1	x	高阻态

(3) 设计原理

原理框图如图 3.12.5 所示。

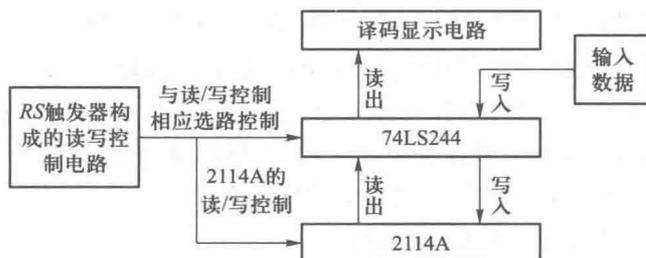


图 3.12.5 原理框图

注:因为是随机存取,所以不必关注 $A_0 \sim A_9$ 地址端的状态, $A_0 \sim A_9$ 可以是随机的,但在读写操作中要保持一致性。

4. 用 2114A 实现静态顺序存取

(1) 4 位二进制同步计数器 74LS163(同步清零)的引脚分布及功能表引脚分布如图 3.12.6 所示,功能如表 3.12.4 所示。

RCO :进位输出端。

CLK :时钟输入端(上升沿有效)。

\overline{CLR} :同步清零输入端(低电平有效)。

ENP :使能端 P。

ENT :使能端 E。

$A \sim D$:并行数据输入端。

\overline{LOAD} :同步并行置入控制端(低电平有效)。

$Q_A \sim Q_D$:输出端。

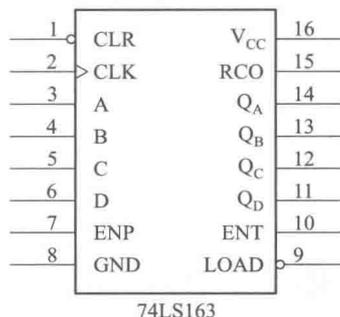


图 3.12.6 74LS163 的引脚分布

表 3.12.4 74LS163 的引脚功能

输入									输出			
\overline{CLR}	\overline{LOAD}	ENP	ENT	CLK	A	B	C	D	Q_A	Q_B	Q_C	Q_D
0	x	x	x	↑	x	x	x	x	0	0	0	0
1	0	x	x	↑	a	b	c	d	a	b	c	d
1	1	1	1	↑	x	x	x	x	计数			
1	1	0	x	x	x	x	x	x	保持			
1	1	x	0	x	x	x	x	x	保持			

(2) 静态顺序存取设计方案

为了实验接线方便,又不影响实验效果,2114A 中地址输入端保留前 4 位(A_0, A_1, A_2, A_3),其余输入端均接地,即顺序存取 2114A 第一行 16 个存储单元的数据。保留的 4 位分别连接同步计数器 74LS163 的 $Q_A \sim Q_D$,通过在 CLK 端加入单脉冲实现地址的递增。初始地址可以通过 CLR 清零,也可以通过 $A \sim D$ 将预置地址从 $Q_A \sim Q_D$ 输入到 2114A。其他部分参考随机存取电路。

三、设计报告要求(设计报告格式见附录 F)

(1) 目的与要求:根据任务书进行填写。

(2) 正文:简单叙述设计过程(包括原理、方案);画出完整的电路原理图,并简述各部分的功能。

(3) 总结与结论:写出设计和调试过程中出现的问题及解决方法;简述心得体会。

第四章

VHDL 语言简介

4.1 VHDL 程序基础

4.1.1 概述

硬件描述语言(HDL)有很多种,但是最流行和通用的只有 VHDL 和 Verilog HDL 两种。它们作为 IEEE 标准化的硬件描述语言,具有许多优点:能够抽象地描述电路结构和行为,支持逻辑设计层次与领域的描述,硬件描述与实现工艺无关,易于理解和可移植性好等。但是 VHDL 相对于 Verilog HDL 而言,在语法上更严谨一些。虽然这样也使它失去了一些灵活性和多样性,但是从文档记录、综合性以及器件和系统级的仿真上讲,VHDL 是一种更好的选择。另外,电路设计的描述层次可分为系统级、算法级、寄存器传输级、门级和晶体管级,VHDL 的建模范围可以从最抽象的系统级一直到门级,从多个层次对电路进行描述和仿真。与 Verilog HDL 相比,VHDL 在门级电路描述方面不如 Verilog HDL,但在系统级抽象描述方面具有很大优势。因此,对于复杂的、大型的系统级设计,使用 VHDL 也更为合适。从学习的角度来说,只要学会了一种硬件描述语言,在使用时可以很方便地转换到另一种。

VHDL 是 Very High Speed Integrated Circuit HDL(超高速集成电路硬件描述语言)的缩写,是在 ADA 语言的基础上发展起来的硬件描述语言。VHDL 是美国国防部在 20 世纪 70 年代末提

出的 VHSIC (Very High Speed Integrated Circuit, 超高速集成电路) 计划的产物, VHSIC 计划的目标是使工业界可以开发比以前更为复杂的集成电路, 加速美国微电子行业的发展。但是由于美国国防部电子系统项目由众多公司承包, 各公司使用自己的语言, 使得设计的可移植性和再开发性很差, 信息交换和设计维护很困难。因此美国国防部委托 IBM 和 Texas Instrument 公司联合开发并在 1981 年提出了一种硬件描述语言 VHSIC Hardware Description Language, 取此项目的名称的第一个字母将这种硬件描述语言命名为 VHDL。

1986 年, 硬件描述语言的 IEEE 标准化组织开始工作, 经过多次讨论和更改, 1987 年 12 月, VHDL 被接纳为标准 HDL, 即 IEEE1076 标准。经过不断更改和完善, 1993 年, VHDL 重新修订并增加了一些功能, 即 IEEE1076-93 标准。严格来说, VHDL-93 并不完全兼容 VHDL-87, 但是只需对后者作简单修改即可成为合法的 VHDL-93 代码。

目前 VHDL 作为 IEEE 的工业标准硬件描述语言, 得到了众多 EDA 公司的支持, 如 Synopsys、Mentor Graphic、Cadence、Altera 等。各公司相继推出了自己的 VHDL 开发环境, 或者宣布自己的开发环境和集成芯片 (IC) 支持 VHDL。因此在电子工程领域, VHDL 已成为事实上的通用硬件描述语言。

VHDL 与其他语言相比, 最大的区别在于设计方法上的差别。VHDL 的主要优点如下。

(1) 与其他的硬件描述语言相比, VHDL 具有更强的行为描述能力, 从而决定了它成为系统设计领域最佳的硬件描述语言之一。强大的行为描述能力是避开具体的器件结构, 从逻辑行为上描述和设计大规模电子系统的重要保证。就目前流行的 EDA 工具和 VHDL 综合器而言, 将基于抽象的行为描述风格的 VHDL 程序综合成为具体的 FPGA 和 CPLD 等目标器件的网表文件已不成问题, 只是在综合与优化效率上略有差异。

(2) VHDL 最初是作为一种仿真标准格式出现的, 因此 VHDL 既是一种硬件电路描述和设计语言, 还是一种仿真语言, 其丰富的仿真语句和库函数, 使得在任何大系统的设计早期 (即尚未完成), 就能用于查验设计系统的功能可行性, 随时可对设计进行仿真模拟。即在远离门级的高层次上进行模拟, 使设计者对整个工程设计的结构和功能的可行性做出决策。

(3) VHDL 语句的行为描述能力和程序结构决定了它具有支持大规模设计的分解和已有设计的再利用功能, 符合市场所需求的大规模系统高效、高速的完成必须由多人甚至多个开发组共同并行工作才能实现的特点。VHDL 中设计实体的概念、程序包的概念、设计库的概念为设计的分解和并行工作提供了有力的支持。

(4) 对于用 VHDL 完成的一个确定的设计, 可以利用 EDA 工具进行逻辑综合和优化, 并自动地把 VHDL 描述设计转变成门级网表。这种方式突破了门级设计的瓶颈, 极大地减少了电路设计的时间和可能发生的错误, 降低了开发成本。应用 EDA 工具的逻辑优化功能, 可以自动地把一个综合后的设计变成一个更高效、更高速的电路系统。反过来, 设计者还可以容易地从综合和优化后的电路获得设计信息, 返回去更新修改 VHDL 设计描述, 使之更为完善。

(5) VHDL 对设计的描述具有相对独立性, 设计者可以不懂硬件的结构, 也可以不考虑最终设计实现的目标器件是什么而进行独立的设计。正因为 VHDL 的硬件描述与具体的工艺技术

和硬件结构无关,VHDL 设计程序的硬件实现目标器件有广阔的选择范围,其中包括各系列的 CPLD、FPGA 及各种门阵列实现目标。

由于 VHDL 具有类属描述语句和子程序调用等功能,对于已完成的设计,在不改变源程序的前提下,只需改变端口类属参量或函数,就能轻易地改变设计的规模和结构。

4.1.2 程序实体 (ENTITY)

VHDL 程序含有实体(entity)、结构体(architecture)、配置(configuration)、包集合(package)、库(library)五部分组成。

简单的程序由实体和结构体两部分组成。实体用于描述设计系统的外部接口信号,结构体用于描述系统的行为、系统数据的流程或者系统组织结构形式。设计实体是 VHDL 程序的基本单元,是电子系统的抽象。

实体作为一个设计实体的组成部分,其功能是对这个设计实体与外部电路进行接口描述。实体是设计实体的表层设计单元,实体说明部分规定了设计单元的输入输出接口信号或引脚,它是设计实体对外的一个通信界面。就一个设计实体而言,外界所看到的仅仅是它的界面上的各种接口。设计实体可以拥有一个或多个结构体,用于描述此设计实体的逻辑结构和逻辑功能,对于外界来说这一部分是不可见的。

不同逻辑功能的设计实体可以拥有相同的实体描述,这是因为实体类似于原理图中的一个部件符号,而其具体的逻辑功能是由设计实体中结构体的描述确定的。实体是 VHDL 的基本设计单元,它可以对一个门电路、一个芯片、一块电路板乃至整个系统进行接口描述。

以下提供一个 2 选 1 多路选择器的设计,使读者能在整体上把握 VHDL 程序设计的结构和特点。

图 4.1.1 为 2 选 1 多路选择器的逻辑图。a 和 b 分别是两个数据输入端的端口名,s 为通道选择控制信号输入端的端口名,y 为输出端的端口名。其逻辑功能可表述为:若 $s = 0$, 则 $y = a$; 若 $s = 1$, 则 $y = b$,VHDL 例程如下。

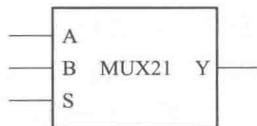


图 4.1.1 2 选 1 多路选择器

【程序 4.1】 2 选 1 多路选择器

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164 ALL;           --IEEE 库使用说明
ENTITY mux21 IS
PORT ( a,b : IN STD_LOGIC;           --器件 mux21 的外部接口信号说明
      s : IN STD_LOGIC;
      y : OUT STD_LOGIC );
END ENTITY mux21;                       --ENTITY 部分称为实体
ARCHITECTURE one OF mux21 IS          --器件 mux21 的内部实现
BEGIN
```

```

y <= a WHEN s = '0' ELSE
b WHEN s = '1' ;
END ARCHITECTURE one;           --ARCHITECTURE 部分成为结构体

```

一、实体的组成

实体由实体名、类属表、端口表、实体说明部分和实体语句部分构成。根据 IEEE 标准, 实体说明单元的一般语句结构如下。

```

ENTITY 实体名 IS
    [ GENERIC(类属表); ]
    [ PORT(端口表); ]
END ENTITY 实体名;

```

实体说明单元必须按照这一结构来编写, 实体应以语句“ENTITY 实体名 IS”开始, 以语句“END ENTITY 实体名;”结束, 其中的实体名可以由设计者自己添加。中间在方括号内的语句描述, 在特定的情况下并非是必需的。例如构建一个 VHDL 仿真测试基准等情况中可以省去方括号中的语句。对于 VHDL 的编译器和综合器来说, 程序文字的大小写是不加区分的, 但为了便于阅读和分辨, 建议将 VHDL 的标识符或基本语句关键词以大写方式表示, 而由设计者添加的内容可以以小写方式来表示。

二、GENERIC 类属说明语句

类属(GENERIC)参量是一种端口界面常数, 常以一种说明的形式放在实体或块结构体前的说明部分。类属为所说明的环境提供了一种静态信息通道。类属与常数不同, 常数只能从设计实体的内部得到赋值, 且不能再改变, 而类属的值可以由设计实体外部提供。因此, 设计者可以从外面通过类属参量的重新设定而容易地改变一个设计实体或一个元件的内部电路结构和规模。

类属说明的一般书写格式如下:

```

GENERIC([常数名:数据类型:设定值] [ ;常数名:数据类型:设定值]...)

```

类属参量以关键词 GENERIC 引导一个类属参量表, 在表中提供时间参数或总线宽度等静态信息。类属表说明用于设计实体和其外部环境通信的参数, 传递静态的信息。类属在所定义的环境中的地位与常数十分接近, 但却能从环境(如设计实体)外部动态地接受赋值, 其行为又有点类似于端口 PORT。因此常如以上的实体定义语句那样将类属说明放在其中且放在端口说明语句的前面。

三、PORT 端口说明

由 PORT 引导的端口说明语句是对一个设计实体界面的说明。其端口表部分对设计实体与外部电路的接口通道进行了说明, 其中包括对每一接口的输入输出模式(MODE 或称端口模式)和数据类型(TYPE)进行了定义。在实体说明的前面, 可以有库的说明, 即由关键词“LIBRARY”和“USE”引导一些对库和程序包使用的说明语句, 其中的一些内容可以为实体端口数据类型的定义所用。

实体端口说明的一般书写格式如下:

PORT(端口名 : 端口模式 数据类型;
 端口名 : 端口模式 数据类型);

其中的端口名是设计者为实体的每一个对外通道所取的名字,端口模式是指这些通道上的数据流动方式,如输入或输出等。数据类型是指端口上流动的数据的表达格式或取值类型,这是由于 VHDL 是一种强类型语言,即对语句中的所有的端口信号、内部信号和操作数的数据类型有严格的规定,只有相同数据类型的端口信号和操作数才能相互作用。IEEE 标准包括以下常用的模式。

1. 输入 (IN)

模式 IN 定义的通道确定为输入端口,并规定为单向只读模式,可以通过此端口将变量 (Variable) 信息或信号 (Signal) 信息读入设计实体中。不用的输入一般接地,以免浮动引入干扰噪声。

2. 输出 (OUT)

模式 OUT 定义的通道确定为输出端口,并规定为单向输出模式,可以通过此端口将信号输出设计实体或者说可以将设计实体中的信号向此端口赋值。一般而言,不用的输出接口不能接地,避免造成输出高电平时烧毁被设计实体。

3. 缓冲 (BUFFER)

模式 BUFFER 的端口与输出模式端口类似,只是缓冲模式允许内部引用该端口的信号。缓冲模式既能用于输出,也能用于反馈。

缓冲不允许双重驱动,不与其他实体的双向端口和输出端口相连。当实体既需要输出又需要反馈时,应采用缓冲端口。

4. 双向模式 (INOUT)

模式 INOUT 定义的通道确定为输入输出双向端口,即从端口的内部看可以对此端口进行赋值,也可以通过此端口读入外部的数据信息;而从端口的外部看,信号既可以从此端口流出,也可以向此端口输入信号。

INOUT 模式包含了 IN、OUT 和 BUFFER 三种模式,因此可替代其中任何一种模式,但为了明确程序中各端口的实际任务一般不作这种替代。

4.1.3 结构体 (ARCHITECTURE)

结构体部分是设计者最主要的工作,用于描述设计的行为和特性。结构体是实体所定义的设计实体中的一个组成部分,结构体描述设计实体的内部结构或外部设计实体端口间的逻辑关系。

结构体将具体实现一个实体。每个实体可以有多个结构体,每个结构体对应着实体不同的结构和算法实现方案,其间的各个结构体的地位是平等的,它们完整地实现了实体的行为。但同一结构体不能为不同的实体所拥有。结构体不能单独存在,它必须有一个界面说明即一个实体。

结构体的一般语言格式如下:

ARCHITECTURE 结构体名 OF 实体名 IS

 [说明语句]

BEGIN

[功能描述语句]

END ARCHITECTURE 结构体名;

在书写格式上实体名必须是所在设计实体的名字,而结构体名可以由设计者自己选择,但当一个实体具有多个结构体时,结构体的取名不可相重。结构体的说明语句部分必须放在关键词“ARCHITECTURE”和“BEGIN”之间。结构体必须以“END ARCHITECTURE 结构体名”作为结束句。

结构体中可以包含五种不同类型的以并行方式工作的语句结构,可以看成是结构体的五个子结构。五种语句结构本身是并行语句,但它们内部所包含的语句并不一定是并行语句,如进程语句内所包含的是顺序语句。五种语句结构分别如下。

(1) 块语句是由一系列并行执行语句构成的组合体,它的功能是将结构体中的并行语句组成一个或多个子模块。

(2) 进程语句定义顺序语句模块,用以将从外部获得的信号值或内部的运算数据向其他的信号进行赋值。

(3) 信号赋值语句将设计实体内的处理结果向定义的信号或界面端口进行赋值。

(4) 子程序调用语句用以调用过程或函数,并将获得的结果赋值于信号。

(5) 元件例化语句对其他的设计实体作元件调用说明,并将此元件的端口与其他的元件信号或高层次实体的界面端口进行连接。

4.2 VHDL 词法基础

4.2.1 标识符

标识符规则是 VHDL 语言中符号书写的一般规则,它不仅对电子系统设计师是个约束,同时也为各种各样的 EDA 工具提供了标准的编译规范。

标识符是最常用的操作符,标识符可以是常数、变量、信号端口、子程序或参数的名字。VHDL 基本标识符的书写遵守如下规则。

(1) 有效的字符,英文字母包括 26 个大小写字母 a~z、A~Z,数字包括 0~9,以及下划线“_”。

(2) 任何标识符必须以英文字母开头。

(3) 必须是单一下划线“_”,且其前后都必须有英文字母或数字。

(4) 标识符中的英语字母不分大小写,VHDL-93 标准还支持扩展标识符。

(5) 扩展标识符以反斜杠来界定,可以以数字打头,如\74LS373\、\Hello World\都是合法的标识符。

(6) 允许包含图形符号(如回车符、换行符等),也允许包含空格符。如\IRDY#\、\C/BE\、\A or B\等都是合法的标识符。

(7) 两个反斜杠之前允许有多个下划线相邻,扩展标识符要分大小写。扩展标识符与短标

标识符不同,扩展标识符如果含有一个反斜杠,则用两个反斜杠来代替它。

下面是合法的标识符:

```
Multi_screen Sig_N Signal0 outPin
```

下面是非法的标识符:

```
Wrong-name wrong%name _wrong_name 2wrong_name wrong_name return
```

VHDL 中保留的关键字如下:

ABS	DOWNTO	LIBRARY	POSTPONED	SRL
ACCESS	FLSE	LINKAGE	PROCEDURE	SUBTYPE
AFTER	FLSIF	LITERAL	PROCESS	THEN
ALIAS	END	LOOP	PUREF	TO
ALL	ENTITY	MAP	RANGE	TRANSPORT
AND	EXIT	MODE	RECORD	TYPE
SLL	FILE	NAND	REGISTER	CONSTANT
ARRAY	FOR	NEW	REECT	UNITES
ASSERT	FUNCTION	NEXT	REM	UNTIL
BEGIN	GENERIC	NOT	RETURN	VARIABLE
BLOCK	GROUP	NULL	ROL	WAIT
BUFFER	IF	ON	SELECT	WHILE
BUS	IMPURE	OPEN	SEVERITY	WITH
CASE	IN	OR	SIGNAL	XNOR
COMPONENT	INERTIAL	OTHERS	SHARED	XOR
STA	INPUT	OUT	CONFIGURATION	SLA
IS	PACKAGE	DISTANT	ARCHITECTURELABEL	
PORT	UNAFFECTED			

4.2.2 VHDL 数据对象

在 VHDL 语言中,凡是赋予一个值的标识都可以叫做对象(object)。VHDL 对象包含专门的数据类型,主要有三种,即变量(VARIABLE)、常量(CONSTANT)和信号(SIGNAL)。

对象说明的一般书写格式为:

对象类别 标识符表:子类型标识[:=初值];

一、变量(VARIABLE)

在 VHDL 语法规则中,变量是一个局部量,只能在进程和子程序中使用。变量不能将信息带出对它作出定义的当前设计单元。变量的赋值是一种理想化的数据传输,是立即发生的,不存在任何延时的行为。VHDL 语言规则不支持变量附加延时语句。变量常用在实现某种算法的赋值语句中。

定义变量的语法格式如下：

VARIABLE 变量名:数据类型:=初始值;

变量赋值语句的语法格式如下：

目标变量名:=表达式;

变量赋值符号是“:=”，变量数值的改变是通过变量赋值来实现的。赋值语句右方的表达式必须是一个与目标变量具有相同数据类型的数值。这个表达式可以是一个运算表达式，也可以是一个数值。通过赋值操作，新的变量值的获得是立刻发生的。变量赋值语句左边的目标变量可以是单值变量，也可以是一个变量的集合即数组型变量。

二、信号(SIGNAL)

信号是电子电路内部硬件实体相互连接的抽象标识。信号通常在结构体、包集合和实体说明中使用，其一般书写格式如下：

SIGNAL 信号名 数据类型 := 初始值;

当信号定义了数据类型和表达方式后，在 VHDL 设计中就能对信号进行赋值了，信号的赋值语句表达式如下：

目标信号名<= 表达式;

这里的表达式可以是一个运算表达式，也可以是数据对象(变量信号或常量)。符号“<=”表示赋值操作，即将数据信息传入。数据信息的传入可以设置延时量，因此目标信号获得传入的数据并不是即时的。

三、常量(CONSTANT)

常量的定义和设置主要是为了使设计实体中的常量更容易阅读和修改。常量是一个恒定不变的值，一旦作了数据类型和赋值定义后，在程序中不能再改变，因而具有全局性意义。

常量定义的一般格式如下：

CONSTANT 常数名数:数据类型:=表达式;

常量的可视性，即常量的使用范围取决于它被定义的位置。如果在程序包中定义，常量具有最大的全局化特征，可以用在调用此程序包的所有设计实体中；常量如果定义在设计实体中，其有效范围为这个实体定义的所有的结构体；如果常量定义在设计实体的某一结构体中，则只能用于此结构体；如果常量定义在结构体的某一单元如一个进程中，则这个常量只能用于这一进程中。这就是常量的可视性规则，这一规则与信号的可视性规则是完全一致的。

4.2.3 VHDL 数据类型

一、标准数据类型

1. 整型(INTEGER)

整数类型的数代表正整数、负整数和零。整数类型与算术整数相似，可以使用预定义的运算操作符，如加“+”、减“-”、乘“*”、除“/”等进行算术运算。在 VHDL 中，整数的取值范围是-2147483647~+2147483647，即可用 32 位有符号的二进制数表示。

整数常量的书写方式示例如下：

15876238	十进制整数
10E6	十进制整数
16#EF#	十六进制整数
8#630#	八进制整数
2#11010010#	二进制整数

2. 实数 (REAL)

在 VHDL 语言中,实数的范围是 $-1.0E+38 \sim +1.0E+38$ 。实数类型仅能在 VHDL 仿真器中使用,VHDL 综合器则不支持实数。因为直接的实数类型的表达和实现相当复杂,目前在电路规模上难以承受。

实数类型的书写方式示例如下：

2.0	十进制浮点数
3.1415926	十进制浮点数
8#123.0#e+4	八进制浮点数

3. 布尔 (BOOLEAN) 数据类型

布尔数据类型实际上是一个二值枚举型数据类型,它的取值只能取 FALSE(伪)和 TRUE(真)。两种综合器将用一个二进制位表示 BOOLEAN 型变量或信号。布尔量不属于数值,因此不能用于运算,它只能通过关系运算符获得。

4. 位 (BIT) 数据类型

位通常用来表示一个信号的值,通常用单引号来括住其位的值,如:

```
TYPE BIT IS( '0' , '1' );
```

位的值'0', '1'代表信号的状态,这与布尔值的真'0',假'1'是不一样的。

5. 位矢量 (BIT_VECTOR) 数据类型

位矢量只是基于 BIT 数据类型的数组,由两个以上的 BIT 的二进制数组成。使用位矢量必须注明位宽,即数组中的元素个数和排列,如:

```
SIGNAL a : BIT_VECTOR(7 DOWNTO 0) ;
```

6. 字符 (CHARACTER) 数据类型

字符类型通常用单引号引起来,如'A'。字符类型区分大小写如'B'不同于'b'。VHDL 语言对大小写英文字母不敏感,但区分字符量中的大小写。字符包括 A~Z、a~z、0~9、空格以及一些特殊字符。

7. 时间 (TIME) 数据类型

VHDL 中唯一的预定义物理类型是时间。完整的时间类型包括整数和物理量单位两部分,整数和单位之间至少留一个空格,如 55 ms,20 ns。时间类型值的范围是整数所定义的范围,为 $-2147483647 \sim +2147483647$ 。

时间类型一般用于仿真,而不用于逻辑综合。

时间单位的描述规范如下:

```
TYPE time IS RANGE -2147483647 TO 2147483647
units
fs ;
ps = 1000 fs ;
ns = 1000 ps ;
us = 1000 ns ;
ms = 1000 us ;
sec = 1000 ms ;
min = 60 sec ;
hr = 60 min ;
end units ;
```

8. 错误等级 (SEVERITY LEVEL)

错误等级常用于电子系统的工作状态。错误等级分为 `note`、`warning`、`error`、`failure`, 即注意、警告、错误、失败四个等级。

错误等级经常用来在仿真器中给开发者提供电子系统的工作状况。

9. 自然数 (NATURAL) 和正整数 (POSITIVE) 数据类型

自然数和正整数都是整数的子类型, 自然数表示非负的整数即零和正整数, 正整数表示整数中非零和非负的数值。

10. 字符串 (STRING) 数据类型

字符串数据类型是字符数据类型的一个非约束型数组, 或称为字符串数组, 字符串必须用双引号标明, 例如: “VHDL”、“STRING”等。

字符串一般用于程序的提示、结果说明等场合。

二、用户自定义数据类型方式

VHDL 允许用户自定义数据类型, 这样就给电子系统设计工程师提供了极大的自由度。用户定义的数据类型的书写规范为:

```
TYPE 数据类型名 IS 数据类型定义;
```

1. 枚举类型

枚举类型, 顾名思义就是把类型中的各个元素列举出来, 方便、直观, 提高了可阅读性。VHDL 中的枚举数据类型是一种特殊的数据类型, 它们是用文字符号来表示一组实际的二进制数。

枚举类型的书写格式为:

```
TYPE 数据类型名 IS (元素, 元素, ...);
```

利用以上规范, 举例如下:

```
TYPE STD_LOGIC IS
```

```
(
    'U'    --未初始化的
    'X'    --强未知的
    '0'    --强 0
    '1'    --强 1
    'Z'    --高阻态
    'W'    --弱未知的
    'L'    --弱 0
    'H'    --弱 1
    '-'    --忽略
);
```

2. 数组类型

数组类型属复合类型,是将一组具有相同数据类型的元素集合在一起作为一个数据对象来处理的数据类型。数组可以是一维(每个元素只有一个下标)数组或多维数组(每个元素有多个下标)。VHDL 仿真器支持多维数组,但 VHDL 综合器只支持一维数组。

数组类型的书写格式如下:

```
TYPE 数组类型名 IS ARRAY 范围 OF 基本类型;
```

3. 记录类型

记录类型与数组类型都属数组,由相同数据类型的对象元素构成的数组称为数组类型的对象,由不同数据类型的对象元素构成的数组称为记录类型的对象。记录是一种异构复合类型,也就是说记录中的元素可以是不同的类型。

构成记录类型的各种不同的数据类型可以是任何一种已定义过的数据类型,也包括数组类型和已定义的记录类型。显然具有记录类型的数据对象的数值是一个复合值,这些复合值是由这个记录类型的元素决定的。

记录类型的书写格式如下:

```
TYPE 记录类型名 IS RECORD
```

```
元素名 : 元素数据类型;
```

```
元素名 : 元素数据类型;
```

```
...
```

```
END RECORD;
```

4.2.4 VHDL 操作符

VHDL 语言的操作符主要有以下 4 种:

- (1) 逻辑运算符(LOGIC OPERATOR);
- (2) 关系运算符(RELATION OPERATOR);

- (3) 算术运算符(ARITHETIC OPERATOR);
- (4) 并置运算(CONCATENATION OPERATOR)。

VHDL 的对象有不同类型,逻辑类型的变量要用逻辑操作符,整数、实数类型的变量要用算术操作符。对于运算操作符和变量类型不匹配的情况,在编译、综合时不予通过。

一、逻辑运算符

在 VHDL 语言中,逻辑运算符有 7 种,分别如下。

- (1) NOT 取反
- (2) AND 与
- (3) OR 或
- (4) NAND 与非
- (5) NOR 或非
- (6) XOR 异或
- (7) XNOR 同或

逻辑运算符适用的变量类型为 STD_LOGIC、BIT、STD_LOGIC_VECTOR,这三种布尔型数据进行逻辑运算时,在方程的左边、右边以及代入的信号类型必须相同。

在一个 VHDL 语句中存在两个逻辑表达式时,左右没有优先级差别。一个逻辑式中,先做括号里的运算,再做括号外的运算。

二、算术运算符

VHDL 语言有 10 种运算符,仅有三种可以被 EDA 工具综合为逻辑电路。VHDL 的算术运算符如下。

- (1) + 加运算
- (2) - 减运算
- (3) * 乘运算
- (4) / 除运算
- (5) MOD 取模运算
- (6) REM 取余运算
- (7) + 正号
- (8) - 负号
- (9) ** 指数运算
- (10) ABS 取绝对值

算数运算符的使用规则如下。

- (1) +、-(正负)操作,即一元运算,可以是整数、实数、物理量。
- (2) +、-(加减)运算符的适用范围可以是整数、实数。而且对于加减运算的两个操作数必须相同。
- (3) *、/(乘除)法的操作数可以同时为整数和实数,物理量乘或除以整数仍为物理量,物

理量除以相同的物理量,商为整数或实数。

(4) MOD、REM(取模、取余)运算的操作符必须是同一类型的数据。

(5) +、-、* (加减乘)能综合为电路,其余运算综合成电路很困难,或者完全是不可能的。

三、关系运算符

关系运算符是两个对象在比较运算时,将两个操作数比较的结果表示出来所使用的操作符,如下。

(1) = 等于

(2) /= 不等于

(3) < 小于

(4) <= 小于等于

(5) > 大于

(6) >= 大于等于

关系运算符在 VHDL 程序设计中有如下规则。

(1) 两个数据比较时,数据类型一定要相同。

(2) =(等于)和/(不等于)适用于所有数据类型对象之间的比较。

(3) 大于、小于、大于等于、小于等于适用于整数、实数、位矢量以及数组类型的比较。

(4) <=符号有两种含义:代入符和小于等于符,要根据上下文判断。

(5) 两个位矢量类型对象比较时,自左向右,按位比较。位矢量在程序包中,对 STD_LOGIC_VECTOR 关系运算专门定义;在位矢量比较时,必须说明调用该包集合。

四、并置运算符

在 VHDL 程序设计中,并置运算符“&”用于位的连接。并置运算符的使用规则如下。

(1) 并置运算符可用于位的连接,形成位矢量。

(2) 并置运算符可用两位矢量的连接构成更大的位矢量。

(3) 位的连接,可以用并置符连接法,也可以用集合体连接法。

五、数值移位运算

如同学习汇编语言一样,当我们将数据乘以 2 或除以 2 时,我们往往会使用移位指令完成,因此在 VHDL 语言中,系统提供了下面 6 个有关 Shift 和 Rotate 的指令。

(1) SRL 逻辑右移

(2) SLL 逻辑左移

(3) SRA 算术右移

(4) SLA 算术左移

(5) ROR 向右旋转

(6) ROL 向左旋转

六、操作符的运算优先级

在 VHDL 程序设计中,逻辑运算、关系运算、算术运算的优先级是不相同的,各种运算的操

4.2.5 注释与数字表示

一、注释

为保证 VHDL 的可读性,在程序行的末尾,双连字符“--”后的文字是对本行程序的注释部分。注释不是设计描述的有效部分,编译后存入数据库的部分不含注释部分。举例如下:

```
Trant<=Frame & idle & irdy & devsel --PCI 总线数据传输译码逻辑之一
```

二、数字

VHDL 语言的数字表示法有十进制表示法、二进制表示法、八进制表示法和十六进制表示法。由于十进制表示法是人们日常生活中默认的数字表示法,所以人们比较熟悉,而二、八、十六进制表示法要加上进制表示符——“基”。这样,数字表示法分为两大类:十进制表示法和基表示法。

1. 十进制数表示法

十进制数书写规范格式为:

十进制文字::=整数[整数][指数]

整数::=数字或下划线连接的数字;

指数::=E[+]整数或 E[-]整数。

整数举例:089,456_78,2E8,5E3...

实数举例:12.0,0.0,3.14159,1.2E-3,2.4E+5...

2. 基表示法

用“基”表示数字的规范书写格式为:

以基表示的数::=基#基于基的整数[基于基的整数]#指数

基::=整数;

基于基的整数::=扩展数字{[下划线]扩展数字},而扩展数字::=数字/字母。

因为十六进制数中,大于9的数字用A、B、C、D、E、F表示,此处数字不再是0~9共10个符号,而是扩展到0~F共16个符号,或者对于前者为扩展数字。

整数举例:十进制值为255的数,用基表示法,写为

2#11111111# 二进制表示法

8#377# 八进制表示法

16#FF# 十六进制表示法

浮点数举例:

2#3 1111-1111#E+4 等于十进制数 63.9375

16#0F#E+1 等于十进制数 0.9375

16#FFFF#E+1 等于十进制数 4095.9375

三、字符、字符串和位串

VHDL 语言中,字符为单括号括起来的 ASCII 符,其规范书写格式为:

字符文字 ::= ‘图形文字’

例如 ‘1’, ‘A’, ‘F’ ...

字符串为双括号括起来的图形字符序列,其规范书写格式为:

字符串文字 ::= “ { 图形字符 } ”

例如“ PCI BUS interface”、“The string contain an information”。

位串是被双括号括起来的扩展数字序列,在数字序列前冠以基数说明符。

位串的规范书写格式为:

位串文字 ::= 基数说明符“位值”

基数说明符如图 4.2.1 所示。

位值 ::= 扩展数字 | 或下划线扩展数字 |

位串的长度是该位串扩展数字序列,等价于二进制数的长度。

举例:

B“1111-1100” 长度为 8 位的二进制数

H“FBC” 长度为 12 等效于 B“1111-1011-1100”

O“371” 长度为 9 等效于 B“011-111-001”

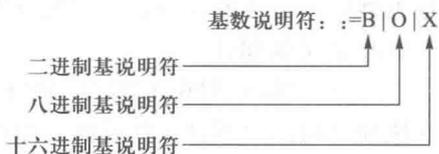


图 4.2.1 基数说明符

4.2.6 预定义属性

VHDL 中预定义属性描述语句有许多实际的应用,可用于对信号或其他项目的多种属性检测或统计。VHDL 中可以具有属性的项目如下。

- (1) 类型、子类型。
- (2) 过程、函数。
- (3) 信号、变量、常量。
- (4) 实体、结构体、配置、程序包。
- (5) 元件。
- (6) 语句标号。

属性是以上各类项目的特性,某一项目的特定属性或特征通常可以用一个值或一个表达式来表示,通过 VHDL 的预定义属性描述语句就可以加以访问。

属性的值与数据对象(信号、变量和常量)的值完全不同,在任一给定的时刻,一个数据对象只能具有一个值,但却可以具有多个属性。VHDL 还允许设计者自己定义属性(即用户定义的属性)。

一、值类属性

1. 值类型属性

值类型属性用来返回类型的边界,有以下 4 种预定义属性。

- (1) T' LEFT: 返回类型或者子类型的左边界。

(2) T' RIGHT: 返回类型或者子类型的右边界。

(3) T' High: 返回类型或者子类型的上限值。

(4) T' Low: 返回类型或者子类型的下限值。

2. 值类数组属性

值类数组属性只有一个即' LENGTH, 该属性返回指定数组范围的总长度, 它用于带某种标量类型的数组范围和带标量类型范围的多维数组。

3. 值类块属性

值类块属性' STRUCTURE 和' BEHAVIOR 返回有关在块和结构体中块是如何建模的信息, 在块和结构体中不含元件具体装配语句, 则属性' BEHAVIOR 将返回真值, 如果块或者结构体中只含元件具体装配语句或被动进程时属性' STRUCTUTE 将返回真值。

二、函数类属性

函数类属性为设计者返回类型、数组和信号信息, 用函数类属性时, 函数调用由输入变元的值返回一个值, 返回值为可枚举值的位置号码、在一个 Δ 时间内信号是否改变的指示或者一个数组的边界。函数类属性可细分为以下 3 个常见的类别。

(1) 函数类型属性, 它返回类型值。

(2) 函数数组属性, 它返回数组的边界。

(3) 函数信号属性, 它返回信号历史信息。

1. 函数类型属性

函数类型属性为有关的类型返回类型内部值的位置号码、返回特定类型输入值的左和右边的值, 函数类型属性分为以下 6 种。

(1) ' POS(value), 返回传入值的位置号码。

(2) ' VAL(value), 返回从该位置号码传入的值。

(3) ' SUCC(value), 返回输入值后类型中的下一个值。

(4) ' PRED(value), 返回输入值前类型中的原先的值。

(5) ' LEFTOF(value), 立即返回一个值到输入值的左边。

(6) ' RIGHTOF(value), 立即返回一个值到输入值的右边。

函数类型属性一般是从可枚举数或物理类型的数转换到整数类型的数。从物理类型转换到整数类型。

2. 函数数组属性

函数数组类属性返回数组类型的边界, 分为以下 4 类。

(1) 数组' LEFT(n); 返回指数范围 n 的左边界。

(2) 数组' RIGHT(n); 返回指数范围 n 的右边界。

(3) 数组' HIGH(n); 返回指数范围 n 的上限值。

(4) 数组' LOW(n); 返回指数范围 n 的下限值。

除了它们用于数组之外, 上升范围时:

数组' LEFT=数组' LOW

数组' RIGHT=数组' HIGH

对下降范围:

数组' LEFT=数组' HIGH

数组' RIGHT=数组' LOW

3. 函数信号属性

函数信号属性用来返回以下有关信号行为功能的信息。

(1) 报告究竟一个信号是否正好有值的变化?

(2) 从上次事件中的跳变过了多少时间?

(3) 该信号原来的值是什么?

有 5 个属性并摘要说明如下。

(1) S' EVENT: 如果当前的 Δ 时间期间发生了事件返回真, 否则返回假(信号是否有值的变化)。

(2) S' ACTIVE: 如果在当前的 Δ 时间期间做了事项处理返回真, 否则返回假(信号是否有值的变化)。

(3) S' LAST_EVENT: 返回从信号原先事件的跳变至今所经历的时间(信号值变了多长时间)。

(4) S' LAST_VALUE: 返回在上一次事件之前 S 的原先值(信号原来的值是什么)。

(5) S' LAST_ACTIVE: 返回自信号原先一次的事项处理至今所经历的时间(信号值变了多长时间)。

三、信号类属性

信号类属性根据另一个信号建一些专用的信号, 由类专用信号为设计者返回有关所附属性的信号信息(在一指定时间范围内该信号是否已经稳定的信息、在信号上有无事项处理的信息和建立的信号的延迟形式)。对这类信号的限制是不能在子程序内部用, 返回的信息和由某种函数属性所提供的功能非常类似, 区别是这类专用信号用于正常信号能用的任何场合, 包括在敏感表中。有如下所示的 4 类属性。

(1) S' DELAYED[(time)]: 建立和参考信号同类型的信号, 该信号后跟参考信号和延时可选时间表示式的时间。

(2) S' STABLE[(time)]: 在选择时间表达式指定的时间内参考信号无事件发生时, 属性建立为真值的布尔信号。

(3) S' QUIET[(time)]: 参考信号或所选时间表达式指定时间内没事项处理时, 属性建立一个为真值的布尔信号。

(4) S' TRANSACTION: 信号上有事件发生或为每个事项处理而翻转它的值时, 该属性建立一个 BIT 类型的信号。

四、类型类的属性

类型类的属性只有一个 t'BASE 类型属性,它必须由另一个值或函数类型属性用该属性。这个属性将返回类型或者子类型的基本类型,这个属性只能作另一属性的前缀。

五、范围类属性

范围类属性返回数组类型的范围值,并由所选的输入参数返回指定的指数范围,这种属性标记如下。

- (1) a' RANGE[(n)]:将返回由参数 n 值指明的第 n 个范围和按指定排序的范围。
- (2) a' REVERSE_RANGE[(n)]:将返回按逆序的范围。

4.3 VHDL 顺序语句结构

VHDL 中的顺序语句与传统的软件编程语言中的语句的执行方式十分相似,语句是按照出现的次序加以执行的。

在 VHDL 中一个进程是由一系列顺序语句构成的,而进程本身属于并行语句,这就是说在同一设计实体中所有的进程是并行执行的,然而任一给定的时刻内在每一个进程内只能执行一条顺序语句。基于行为仿真一个进程与其设计实体的其他部分进行数据交换的方式只能通过信号或端口。如果要在进程中完成某些特定的算法和逻辑操作也可以通过依次调用子程序来实现,但子程序本身并无顺序和并行语句之分。利用顺序语句可以描述逻辑系统中的组合逻辑、时序逻辑或它们的综合体。

VHDL 有如下 6 类基本顺序语句:

- (1) 赋值语句;
- (2) 流程控制语句;
- (3) 等待语句;
- (4) 子程序调用语句;
- (5) 返回语句;
- (6) 空操作语句。

4.3.1 赋值语句

赋值语句有两种,即信号赋值语句和变量赋值语句。赋值符号只有两种:信号赋值符号是“<=”,变量赋值符号是“:=”。

变量赋值语句和信号赋值语句的语法格式如下:

变量赋值目标:= 赋值源

信号赋值目标<= 赋值源

变量赋值与信号赋值的区别在于变量具有局部特征,它的有效性只局限于所定义的一个进

程中或一个子程序中,它是一个局部的暂时性数据对象(在某些情况下),对于它的赋值是立即发生的(假设进程已启动),即是一种时间延迟为零的赋值行为。

信号则不同,信号具有全局性特征,它不但可以作为一个设计实体内部各单元之间数据传送的载体,而且可通过信号与其他的实体进行通信(端口本质上也是一种信号)。信号的赋值并不是立即发生的,它发生在一个进程结束时,赋值过程总是有某种延时的,它反映了硬件系统的重要特性。

读者可以从程序 4.2 看出信号与变量赋值的特点及它们的区别。当同一赋值目标处于不同进程中时,其赋值结果就比较复杂了,这可以看成是多个信号驱动源连接在一起,可以发生线或、线或、三态等不同结果。

【程序 4.2】 赋值语句

```
SIGNAL s1,s2 : STD_LOGIC;
SIGNAL svec : STD_LOGIC_VECTOR (0 TO 7);
...
PROCESS (s1,s2)
VARIABLE v1,v2 : STD_LOGIC;
BEGIN
v1 := '1' ;           --立即将 v1 置位为 1
v2 := '1' ;           --立即将 v2 置位为 1
s1 <= '1' ;           --s1 被赋值为 1
s2 <= '1' ;           --s2 被赋值为 1
svec(0) <= v1;        --将 v1 在上面的赋值 1 赋给 svec(0)
svec(1) <= v2;        --将 v2 在上面的赋值 1 赋给 svec(1)
svec(2) <= s1;        --将 s1 在上面的赋值 1 赋给 svec(2)
svec(3) <= s2;        --将最下面的赋予 s2 的值'0' 赋给 svec(3)
v1 := '0' ;           --将 v1 置入新值 0
v2 := '0' ;           --将 v2 置入新值 0
s2 <= '0' ;           --由于这是 s2 最后一次赋值,赋值有效
svec(4) <= v1;        --将 v1 在上面的赋值 0 赋给 svec(4)
svec(5) <= v2;        --将 v2 在上面的赋值 0 赋给 svec(5)
svec(6) <= s1;        --将 s1 在上面的赋值 1 赋给 svec(6)
svec(7) <= s2;        --将 s2 在上面的赋值 0 赋给 svec(7)
END PROCESS ;
```

4.3.2 流程控制语句

流程控制语句通过条件控制开关决定是否执行一条或几条语句或重复执行一条或几条语句

或跳过一条或几条语句。流程控制语句共有以下 5 种：

- (1) IF 语句；
- (2) CASE 语句；
- (3) LOOP 语句；
- (4) NEXT 语句；
- (5) EXIT 语句。

一、IF 语句

IF 语句是一种条件语句，它根据语句中所设置的一种或多种条件有选择地执行指定的顺序语句。IF 语句的语句结构有以下 3 种。

1. 单 IF 语句

IF 条件句 Then

 顺序语句

END IF

2. 二选择的 IF 语句

IF 条件句 Then

 顺序语句

ELSE

 顺序语句

END IF

3. 多选择的 IF 语句

IF 条件句 1 Then

 顺序语句 1

ELSIF 条件句 2 Then

 顺序语句 2

...

ELSE

 顺序语句

END IF;

【程序 4.3】 IF 语句实例

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164 ALL;
ENTITY control_stmts IS
    PORT (a, b, c: IN BOOLEAN;
          output: OUT BOOLEAN);
END control_stmts;
```

```

ARCHITECTURE example OF control_stmts IS
BEGIN
    PROCESS (a, b, c)
        VARIABLE n: BOOLEAN;
    BEGIN
        IF a THEN
            n := b;
        ELSE
            n := c;
        END IF;
        output <= n;
    END PROCESS;
END example;

```

程序 4.3 对应的硬件电路如图 4.3.1 所示,这是一个多路通道选择器, a 是通道控制信号。

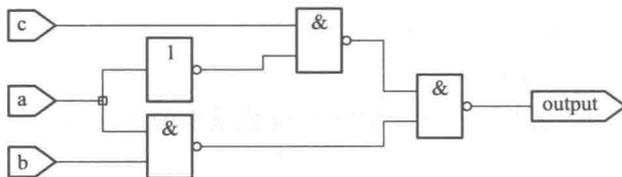


图 4.3.1 程序 4.3 的硬件实现电路

二、CASE 语句

CASE 语句根据满足的条件直接选择多项顺序语句中的一项执行。

CASE 语句的结构如下:

CASE 表达式 IS

When 选择值 1 => 顺序语句;

When 选择值 2 => 顺序语句;

...

When others => 顺序语句;

END CASE;

当执行到 CASE 语句时,首先计算表达式的值,然后根据条件句中与之相同的选择值,执行对应的顺序语句最后结束。CASE 语句表达式可以是一个整数类型或枚举类型的值,也可以是由这些数据类型的值构成的数组。

【程序 4.4】 4 选 1 的多路选择器

```

LIBRARY IEEE

```

```

USE IEEE.STD_LOGIC_1164.ALL
ENTITY mux41 IS
PORT (s1, s2 : IN STD_LOGIC;
      a, b, c, d : IN STD_LOGIC;
      z : OUT STD_LOGIC);
END ENTITY mux41
ARCHITECTURE activ OF mux41 IS
    SIGNAL s : STD_LOGIC_VECTOR (1 DOWNTO 0);
BEGIN
    s <= s1 & s2 ;
    PROCESS (s , a, b, c, d)           --这里以 s 为敏感信号而非 s1 和 s2
    BEGIN
        CASE s IS
            WHEN "00" => z <= a ;
            WHEN "01" => z <= b ;
            WHEN "10" => z <= c ;
            WHEN "11" => z <= d ;
            WHEN OTHERS => z <= 'X' ; --注意这里的 X 必须大写
        END CASE
    END PROCESS
END activ;

```

注意程序 4.4 中的第五个条件句是必需的,因为对于定义为 STD_LOGIC_VECTOR 数据类型的 s,在 VHDL 综合过程中它可能的选择值除了 00,01,10 和 11 外还可以有其他定义于 STD_LOGIC 的选择值。此例的逻辑图如图 4.3.2 所示。另外需要特别注意 WHEN OTHERS => z <= 'X',一句中的 X 必须大写,否则为错。这是由于必须与程序包中对数据类型 STD_LOGIC 的最初定义一致。

三、LOOP 语句

LOOP 语句就是循环语句,它可以使所包含的一组顺序语句被循环执行,其执行次数可由设定的循环参数决定。

FOR_LOOP 语句语法格式如下:

```

[LOOP 标号] FOR 循环变量 IN 循环次数范围 LOOP
    顺序语句;
END LOOP [LOOP 标号];

```

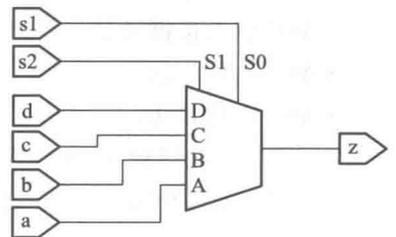


图 4.3.2 4 选 1 多路选择器

WHILE_LOOP 语句语法格式如下：

```
[LOOP 标号] WHILE 循环控制条件 LOOP
    顺序语句；
END LOOP [标号]；
```

与 FOR_LOOP 语句不同的是 WHILE_LOOP 语句并没有给出循环次数范围,没有自动递增循环变量的功能,而是只给出了循环执行顺序语句的条件。这里的循环控制条件可以是任何布尔表达式,如 $a=0$ 或 $a>b$ 。当条件为 TRUE 时,继续循环;为 FALSE 时,跳出循环,执行 END LOOP 后的语句。

四、NEXT 语句

NEXT 语句主要用在 LOOP 语句执行中进行有条件的或无条件的转向控制。

NEXT 语句语法格式如下：

```
NEXT [LOOP 标号] [WHEN 条件表达式]；
```

若 LOOP 标号缺省,则 NEXT 语句作用于当前最内层的循环;否则转到指定的循环中。若 WHEN 子句出现但条件为 False,则循环正常继续。

五、EXIT 语句

EXIT 语句与 NEXT 语句具有十分相似的语句格式和转跳功能,它们都是 LOOP 语句的内部循环控制语句。

EXIT 语句语法格式如下：

```
EXIT [LOOP 标号] [WHEN 条件表达式]；
```

EXIT 语句有条件或者无条件的终止当前循环迭代并终止该循环。若 LOOP 标号缺省,则 EXIT 语句用于当前最内层的循环,否则转到指定的循环中。若 WHEN 子句出现,但条件为 FALSE,则循环正常继续。

4.3.3 WAIT 语句

在进程中(包括过程中)当执行到 WAIT 等待语句时运行程序将被挂起(Suspension),直到满足此语句设置的结束挂起条件后,将重新开始执行进程或过程中的程序。

对于不同的结束挂起条件的设置,WAIT 语句的基本格式有 3 种：

WAIT ON 信号表；

WAIT UNTIL 条件表达式；

WAIT FOR 时间表达式；

【程序 4.5】

```
SIGNAL s1,s2 : STD_LOGIC;
```

```
...
```

```
PROCESS
```

```
BEGIN
```

```
...
    WAIT ON s1,s2;
END PROCESS;
```

在执行了此例中所有的语句后,进程将在 WAIT 语句处被挂起一直到 s1 或 s2 中任一信号发生改变时,进程才重新开始。此例中的 PROCESS 语句未列出任何敏感量,VHDL 规定已列出敏感量的进程中不能使用任何形式的 WAIT 语句。一般地,WAIT 语句可用于进程中的任何地方。

4.3.4 NULL 语句

空操作语句的语句格式如下:

```
NULL;
```

空操作语句不完成任何操作,它唯一的功能就是使逻辑运行流程跨入下一步语句的执行。NULL 常用于 CASE 语句中为满足所有可能的条件,利用 NULL 来表示所余的不用条件下的操作行为。

4.3.5 RETURN 语句

RETURN 语句只能用在函数与过程体内,并用来结束当前最内层函数或过程体的执行,其基本格式如下:

```
RETURN [表达式];
```

过程体中的 RETURN 语句一定不能有表达式,而函数体中的 RETURN 语句必须有一个表达式。它也是结束函数体唯一的执行条件。函数不能走到 END 的位置,而必须经过 RETURN 语句。

4.3.6 过程调用语句

过程调用就是执行一个给定名字和参数的过程。调用过程的语句格式如下:

```
过程名([ [形参名=> ]实参表达式 | [形参名=> ]实参表达式 ));
```

一个过程的调用将分别完成以下 3 个步骤。

- (1) 首先将 IN 和 INOUT 模式的实参值赋给欲调用的过程中与它们对应的形参。
- (2) 然后执行这个过程。
- (3) 最后将过程中 IN 和 INOUT 模式的形参值赋还给对应的实参。

4.4 VHDL 并行语句结构

并行语句结构是最具硬件描述语言特色的,在 VHDL 中并行语句有多种语句格式,各种并

行语句在结构体中的执行是同步进行的,或者说是并行运行的,其执行方式与书写的顺序无关。每个并行语句是一个功能单元,各个功能单元组成一个结构体,其结构如图 4.4.1 所示。

VHDL 语言中的并行语句主要有以下 7 种。

(1) 并行信号赋值语句(Concurrent Signal Assignments)。

(2) 进程语句(Process Statements)。

(3) 块语句(Block Statements)。

(4) 条件信号赋值语句(Selected Signal Assignments)。

(5) 元件例化语句(Component Instantiations),其中包括类属配置语句。

(6) 生成语句(Generate Statements)。

(7) 并行过程调用语句(Concurrent Procedure Calls)。

并行语句在结构体中的使用格式如下:

ARCHITECTURE 结构体名 OF 实体名 IS

说明语句

BEGIN

并行语句

END ARCHITECTURE 结构体名;

4.4.1 进程语句

进程语句是并行处理语句,即各个进程是同时处理的,在结构体中多个 PROCESS 同时并发运行。PROCESS 语句在 VHDL 中是描述硬件并行工作行为最常用、最基本的语句。

PROCESS 语句有如下特点。

(1) 进程内部所有语句是顺序执行的。

(2) 多进程之间是并行执行,并可访问结构体或实体中所定义的信号。

(3) 进程的启动是由进程标识符 PROCESS 后的敏感信号表所标明的信号来触发的,也可以用 WAIT 语句等待一个触发条件的成立。

(4) 各进程之间的通信是由信号来传递的。

PROCESS 的基本语句结构如下:

[进程标号] PROCESS [(敏感信号参数表)]

变量说明语句;

BEGIN

顺序描述语句;

END PROCESS [进程标号];

进程语句中的敏感信号表和 WAIT 语句的作用一致,都是进程启动、触发的条件。为不产生

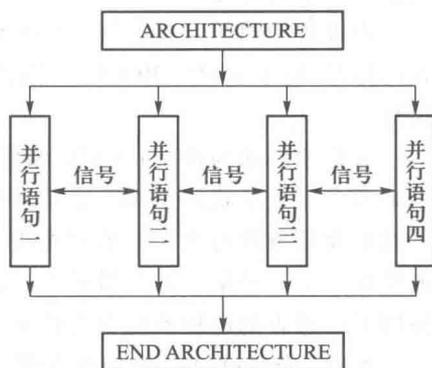


图 4.4.1 并行语句的结构关系

两个进程启动条件,避免使进程产生误触发,在进程语句中,信号敏感表和 WAIT 语句不能同时共存于一个进程之中。

由进程的组织结构可知,PROCESS 不是单条语句,而是代表某一功能的程序段。它起于进程的标号,终于 END PROCESS,描述了一个电子硬件模块的工作进程,而且可以反复触发、反复执行。

进程的启动与执行:进程的激活必须由敏感信号表中定义的任一敏感信号的变化来启动,否则必须有一个显式的 WAIT 语句来激励。这就是说进程既可以通过敏感信号的变化来启动,也可以由满足条件的 WAIT 语句而激活。因此进程中必须定义显式或隐式的敏感信号,如果一个进程对一个信号集合总是敏感的,那么我们可以使用敏感表来指定进程的敏感信号,但是在一个使用了敏感表的进程不能含有任何 WAIT 语句。

WAIT 语句:WAIT 语句在进程中起到与敏感信号一样重要的作用。敏感信号触发进程的执行,WAIT 语句同步进程的执行。

WAIT 语句的格式为:

WAIT	--无限等待
WAIT ON	--敏感信号量
WAIT UNTIL 表达式	--表达式成立时进程启动
WAIT FOR 时间表达式	--时间到,进程启动

程序 4.6 有一个产生组合电路的进程,它描述了一个十进制加法器。对于每 4 位输入 in1(3 DOWNTO 0),此进程对其作加 1 操作,并将结果由 out1(3 DOWNTO 0)输出。由于是加 1 组合电路故无记忆功能。程序 4.6 综合后的硬件电路图如图 4.4.2 所示。

【程序 4.6】十进制加法器

```

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164 ALL;
USE IEEE.STD_LOGIC_UNSIGNED ALL;
ENTITY cnt10 IS
PORT(clr : IN STD_LOGIC;
      in1 : IN STD_LOGIC_VECTOR(3 DOWNTO 0);
      out1 : OUT STD_LOGIC_VECTOR(3 DOWNTO 0) );
END cnt10
ARCHITECTURE actv OF cnt10 IS
BEGIN
PROCESS(in1,clr)
BEGIN
IF (clr = '1' OR in1 = "1001") THEN
out1 <= "0000";          --有清零信号或计数已达 9 out1 输出 0

```

```

ELSE          --否则作加 1 操作
out1 <= in1+1;  --注意:使用了重载算符“+”
END IF       --重载算符“+”是在库 STD_LOGIC_UNSIGNED 中预先声明的
END PROCESS;
END actv;

```

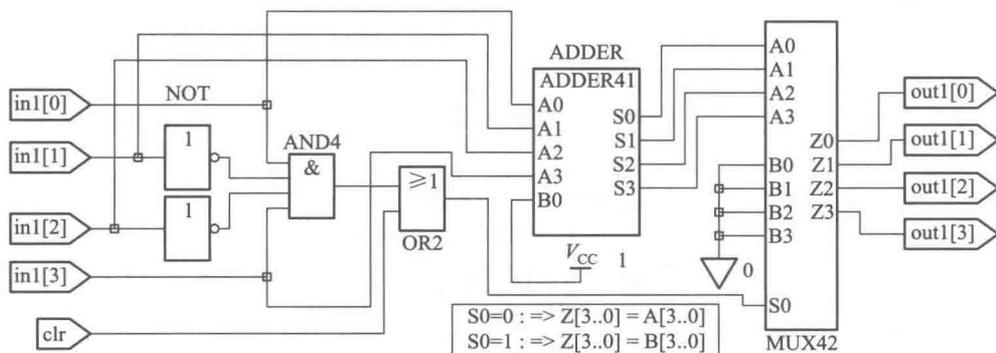


图 4.4.2 程序 4.6 综合后的硬件电路图

4.4.2 块语句

BLOCK 语句是结构体中积木化设计语言,适用于复杂的项目设计。块语句本身是并行语句结构,而且它的内部也都是由并行语句构成的。与其他的并行语句相比,块语句本身并没有独特的功能,它只是一种并行语句的组合方式,利用它可以将程序编排得更加清晰更有层次。

BLOCK 是 VHDL 中具有的一种划分机制,这种机制允许设计者合理地将一个模块分为数个区域,在每个块都能对其局部信号、数据类型和常量加以描述和定义。任何能在结构体的说明部分进行说明的对象都能在 BLOCK 说明部分中进行说明。

BLOCK 语句的表达格式如下:

块标号 BLOCK [块保护表达式]

接口说明;

类属说明;

BEGIN

并行语句

END BLOCK 块标号;

端口间的信号传递和端口映射。

BLOCK 语句是可以嵌套的。内层 BLOCK 块可以使用外层 BLOCK 所定义的信号,但外层 BLOCK 块不能使用内层 BLOCK 块定义的信号。

BLOCK 块是一个独立的子结构,可以包含 PORT 语句、GENERIC 语句,允许设计者通过这

两个语句将 BLOCK 块内的信号变化传给 BLOCK 外的信号。同样,也可以将 BLOCK 块外部的信号变化传递给 BLOCK 内部信号。

PORT MAP 语句和 GENERIC 语句允许在一个新的设计中使用 BLOCK 块。新设计的块完成新的功能。如果 PORT 端口名和 GENERIC 参数不一致,在模块中采用 PORT MAP 和 GENERIC 映射就能解决这个问题。

【程序 4.7】

```

...
b1 : BLOCK                --定义块 b1
    SIGNAL s : BIT ;      --在 b1 块中定义 s
    BEGIN
        S <= a AND b ;    --向 b1 中的 s 赋值
b2 : BLOCK                --定义块 b2 套于 b1 块中
    SIGNAL s : BIT ;      --定义 b2 块中的信号 s
    BEGIN
        s <= c AND d ;    --向 b2 中的 s 赋值
b3 : BLOCK
    BEGIN
        z <= s ;          --此 s 来自 b2 块
END BLOCK b3 ;
END BLOCK b2 ;
y <= s ;                  --此 s 来自 b1 块
END BLOCK b1 ;

```

程序 4.7 是对嵌套块的语法现象作一些说明,实际描述的是如图 4.4.3 所示的两个相互独立的 2 输入与门。

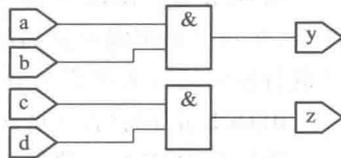


图 4.4.3 两个 2 输入与门

4.4.3 并行信号赋值语句

并行信号赋值语句有以下 3 种形式。

- (1) 简单信号赋值语句。
- (2) 条件信号赋值语句。
- (3) 选择信号赋值语句。

这三种信号赋值语句的共同点是赋值目标必须都是信号,所有赋值语句与其他并行语句一样,在结构体内的执行是同时发生的,与它们的书写顺序无关。

一、简单信号赋值语句

并行简单信号赋值语句是 VHDL 并行语句结构的最基本的单元。它的语句格式如下:

赋值目标 <= 表达式;

式中赋值目标的数据对象必须是信号,它的数据类型必须与赋值符号右边表达式的数据类型一致。

程序 4.8 中的五条赋值语句是并行操作的。

【程序 4.8】

```
ARCHITECTURE curt OF bc1 IS
    SIGNAL s1 : STD_LOGIC ;
BEGIN
    output1 <= a AND b ;
    output2 <= c + d ;
    B1 : BLOCK
        SIGNAL e, f, g, h : STD_LOGIC ;
    BEGIN
        g <= e OR f ;
        h <= e XOR f ;
    END BLOCK B1
    s1 <= g ;
END ARCHITECTURE curt;
```

二、条件信号赋值语句

```
赋值目标 <= 表达式 WHEN 赋值条件 ELSE
    表达式 WHEN 赋值条件 ELSE
    ...
    表达式;
```

在结构体中的条件信号赋值语句的功能与在进程中的 IF 语句相同,在执行条件信号语句时,每一赋值条件是按书写的先后关系逐项测定的,一旦发现(赋值条件=TRUE)立即将表达式的值赋给赋值目标变量。

【程序 4.9】

```
...
z <= a WHEN p1 = '1' ELSE
    b WHEN p2 = '1' ELSE
    c ;
...
```

由于条件测试的顺序性,第一子句具有最高赋值优先级,第二句其次,第三句最后。这就是说如果当 p1 和 p2 同时为 1 时,z 获得的赋值是 a。

三、选择信号赋值语句

选择信号赋值语句的语句格式如下:

WITH 选择表达式 SELECT

```

赋值目标信号 <= 表达式 WHEN 选择值,
                表达式 WHEN 选择值,
                ...
                表达式 WHEN 选择值;

```

选择信号赋值语句本身不能在进程中应用,但其功能却与进程中的 CASE 语句的功能相似。CASE 语句的执行依赖进程中敏感信号的改变而启动进程,而且要求 CASE 语句中各子句的条件不能有重叠,必须包容所有的条件。

程序 4.10 是一个简化的指令译码器,如图 4.4.4 所示,对应于由 a、b、c 三个位构成的不同指令码,由 data1 和 data2 输入的两个值将进行不同的逻辑操作,并将结果从 dataout 输出,当不满足所列的指令码时,将输出高阻态。

【程序 4.10】

```

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY decoder IS
    PORT ( a b c : IN STD_LOGIC;
          data1 data2 : IN STD_LOGIC;
          dataout : OUT STD_LOGIC );
END decoder;
ARCHITECTURE concunt OF decoder IS
    SIGNAL instruction : STD_LOGIC_VECTOR(2 DOWNTO 0)
BEGIN
    instruction <= c & b & a ;
    WITH instruction SELECT
    dataout <= data1 AND data2 WHEN "000"
              data1 OR data2 WHEN "001"
              data1 NAND data2 WHEN "010"
              data1 NOR data2 WHEN "011"
              data1 XOR data2 WHEN "100"
              data1 XNOR data2 WHEN "101"
              'Z' WHEN OTHERS ;
END concunt ;

```

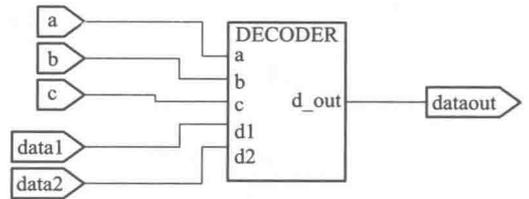


图 4.4.4 程序 4.10 的指令译码器

4.4.4 过程调用语句

过程调用语句可以作为一个并行语句直接出现在结构体或块语句中。并行过程调用语句的功能等效于包含了同一个过程调用语句的进程。

过程的语句格式是：

```
PROCEDURE 过程名(参数表) IS
```

```
    [说明部分;]
```

```
BEGIN
```

```
    顺序语句;
```

```
END 过程名;
```

参数表可以对常数、变量和信号三类数据对象目标作出说明,并用关键词 IN、OUT 和 INOUT 定义这些参数的工作模式,即信息的流向。如果没有指定模式,则默认为 IN。

程序 4.11 中首先设计了一个过程 check,用于确定一给定位宽的位矢是否只有一位是 1,如果不是,则将 check 中的输出参量 error 设置为 TRUE 布尔量。

【程序 4.11】

```
PROCEDURE check(SIGNAL a : IN STD_LOGIC_VECTOR;      --在调用时
                SIGNAL error : OUT BOOLEAN ) IS      --再定位宽
    VARIABLE found_one : BOOLEAN := FALSE ;          --设初始值
BEGIN
    FOR i IN a'RANGE LOOP                            --对位矢量 a 的所有的位元素进行循环检测
        IF a(i) = '1' THEN                            --发现 a 中有 '1'
            IF found_one THEN                          --若 found_one 为 TRUE 则表明发现了一个以上的
                '1'
                error<=TRUE;                          --发现了一个以上的'1' 令 error 为 TRUE
            RETURN;                                    --结束过程
            END IF;
            Found_one :=TRUE;                          --在 a 中已发现了一个'1'
            End IF;
        End LOOP;                                     --再测 a 中的其他位
        error<=NOT found_one;                          --如果没有任何'1' 被发现,error 将被置 TRUE
    END PROCEDURE check
```

程序 4.12 是对不同位宽的位矢信号利用以上的过程进行检测的并行过程调用程序。块 CHBLK 的逻辑电路结构图如图 4.4.5 所示。

【程序 4.12】

...

CHBLK BLOCK

```
SIGNAL s1: STD_LOGIC_VECTOR (0 TO 0);    --过程调用前设定位矢尺寸
SIGNAL s2: STD_LOGIC_VECTOR (0 TO 1);
SIGNAL s3: STD_LOGIC_VECTOR (0 TO 2);
SIGNAL s4: STD_LOGIC_VECTOR (0 TO 3);
SIGNAL e1, e2, e3, e4: Boolean;
```

BEGIN

```
Check (s1, e1);    --并行过程调用 关联参数名为 s1 e1
Check (s2, e2);    --并行过程调用 关联参数名为 s2 e2
Check (s3, e3);    --并行过程调用 关联参数名为 s3 e3
Check (s4, e4);    --并行过程调用 关联参数名为 s4 e4
```

END BLOCK;

...

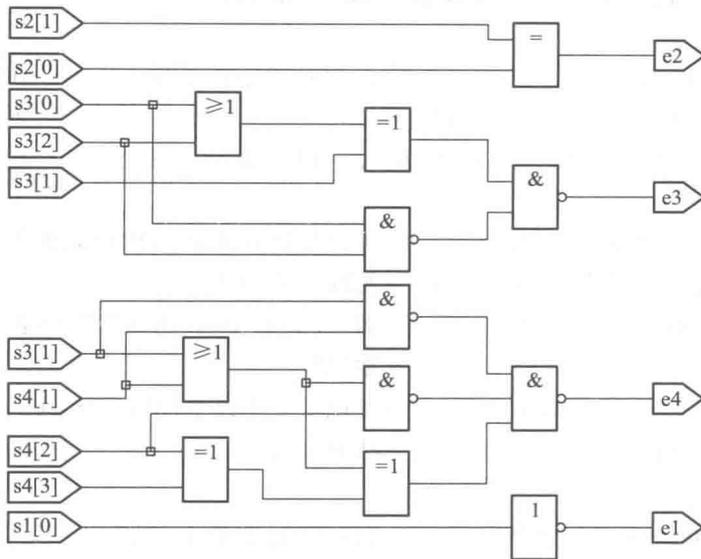


图 4.4.5 块 CHBLK 的逻辑电路结构图

4.4.5 元件例化语句

元件例化就是引入一种连接关系,将预先设计好的设计实体定义为一个元件,然后利用特定的语句将此元件与当前的设计实体中的指定端口相连接,从而为当前设计实体引入一个新的低一级的设计层次。元件例化是使 VHDL 设计实体构成自上而下层次化设计的一种重要途径。

元件例化语句由两部分组成,前一部分是对一个现成的设计实体定义为一个元件,第二部分则是此元件与当前设计实体中的连接说明,它们的语句格式如下:

第一部分:定义元件

COMPONENT 元件名

 GENERIC 说明语句;

 PORT 说明语句;

END COMPONENT;

第二部分:调用元件

例化名:元件名 PORT MAP([端口名=>] 连接端口名,...);

元件例化语句中所定义的元件的端口名与当前系统的连接端口名的接口表达有两种方式:一种是名字关联方式,在这种关联方式下,例化元件的端口名和关联连接符号“=>”两者都是必须存在的。这时,端口名与连接端口名的对应式,在 PORT MAP 句中的位置可以是任意的。另一种是位置关联方式,若使用这种方式,端口名和关联连接符号都可省去,在 PORT MAP 子句中只要列出当前系统中的连接端口名就行了,但要求连接端口名的排列方式与所需例化的元件端口定义中的端口名一一对应。

4.4.6 类属映射语句

GENERIC 语句用于不同层次设计模块之间信息的传递和参数的传递,可用于位矢量的长度、数组的位长、器件的延时时间等参数的传递。GENERIC 可用于设计从外部端口改变元件内部参数或结构规模的元件,或称类属元件,这些元件在例化中特别方便,在改变电路结构或元件升级方面显得尤为便捷。其语句格式如下:

GENERIC(信号定义列表);

4.4.7 生成语句

GENERATE 语句用来产生多个相同的结构和描述规则结构,如块阵列、元件例化或进程。GENERIC 语句的语句格式有如下两种形式:

标号 FOR 循环变量 IN 取值范围 GENERATE

 并行语句;

 END GENERATE [标号名];

标号 IF 条件 GENERATE

 并行语句;

 END GENERATE [标号名];

FOR 语句结构主要是用来描述设计中的一些有规律的单元结构,其生成参数及其取值范围的含义和运行方式与 LOOP 语句十分相似,但需注意,从软件运行的角度上看,FOR 语句格式中生成参数(循环变量)的递增方式具有顺序的性质,但从最后生成的设计结构却是完全并行的,

这就是为什么必须用并行语句来作为生成设计单元的缘故。IF_GENERATE 语句主要用在电路两端的不规则部分。

4.4.8 库 LIBRARY

在利用 VHDL 进行工程设计中,为了提高设计效率以及使设计遵循某些统一的语言标准或数据格式,有必要将一些有用的信息汇集在一个或几个库中以供调用。这些信息可以是预先定义好的数据类型、子程序等设计单元的集合体(程序包),或预先设计好的各种设计实体(元件库程序包)。因此可以把库看成是一种用来存储预先完成的程序包、数据集合体和元件的仓库。

通常库中放置不同数量的程序包,而程序包中又可放置不同数量的子程序;子程序中又含有函数、过程、设计实体元件等基础设计单元。

库 LIBRARY 的语句格式如下:

LIBRARY 库名;

这一语句即相当于为其后的设计实体打开了以此库名命名的库,以便设计实体可以利用其中的程序包。

一、VHDL 常用库种类

VHDL 程序设计中常用的库有以下几种。

1. IEEE 库

IEEE 库是 VHDL 设计中最为常见的库,它包含有 IEEE 标准的程序包和其他一些支持工业标准的程序包。IEEE 库中的标准程序包主要包括 STD_LOGIC_1164、NUMERIC_BIT 和 NUMERIC_STD 等程序包。其中的 STD_LOGIC_1164 是最重要和最常用的程序包,大部分基于数字系统设计的程序包都是以此程序包中设定的标准为基础的。

此外还有一些程序包虽非 IEEE 标准,但由于其已成事实上的工业标准也都并入了 IEEE 库。这些程序包中最常用的是 Synopsys 公司的 STD_LOGIC_ARITH、STD_LOGIC_SIGNED 和 STD_LOGIC_UNSIGNED 程序包。目前流行于我国的大多数 EDA 工具都支持 Synopsys 公司的程序包。一般基于大规模可编程逻辑器件的数字系统设计,IEEE 库中的四个程序包 STD_LOGIC_1164、STD_LOGIC_ARITH、STD_LOGIC_SIGNED 和 STD_LOGIC_UNSIGNED 已足够使用。另外需要注意的是在 IEEE 库中符合 IEEE 标准的程序包并非符合 VHDL 语言标准,如 TD_LOGIC_116 程序包。因此在使用 VHDL 设计实体的前面必须以显式表达出来。

2. STD 库

VHDL 语言标准定义了两个标准程序包,即 STANDARD(标准程序包)和 TEXTIO(文件输入/输出程序包),它们都被收入在 STD 库中,只要在 VHDL 应用环境中即可随时调用这两个程序包中的所有内容,即在编译和综合过程中,VHDL 的每一项设计都自动地将其包含进去了。由于 STD 库符合 VHDL 语言标准,在应用中不必如 IEEE 库那样以显式表达出来。

3. WORK 库

WORK 库是用户的 VHDL 设计的现行工作库,用于存放用户设计和定义的一些设计单元和程序包,因而是用户的临时仓库,用户设计项目的成品、半成品模块,以及先期已设计好的元件都放在其中。WORK 库自动满足 VHDL 语言标准,在实际调用中也不必以显式预先说明。基于 VHDL 所要求的 WORK 库的基本概念,在 PC 机或工作站上利用 VHDL 进行项目设计,不允许在根目录下进行,而是必须为此设定一个目录,用于保存所有此项目的设计文件,VHDL 综合器将此目录默认为 WORK 库。但必须注意,工作库并不是这个目录的目录名,而是一个逻辑名,综合器将指示器指向该目录的路径。VHDL 标准规定工作库总是可见的,因此,不必在 VHDL 程序中明确指定。

4. VITAL 库

使用 VITAL 库,可以提高 VHDL 门级时序模拟的精度,因而只在 VHDL 仿真器中使用。库中包含时序程序包 VITAL_TIMING 和 VITAL_PRIMITIVES。VITAL 程序包已经成为 IEEE 标准,在当前的 VHDL 仿真器的库中,VITAL 库中的程序包都已经并到 IEEE 库中。实际上由于各 FPGA/CPLD 生产厂商的适配工具都能为各自的芯片生成带时序信息的 VHDL 门级网表,用 VHDL 仿真器仿真该网表可以得到非常精确的时序仿真结果。因此,基于实用的观点,在 FPGA/CPLD 设计开发过程中,一般并不需要 VITAL 库中的程序包。

除了以上提到的库外,EDA 工具开发商为了 FPGA/CPLD 开发设计上的方便,都有自己的扩展库和相应的程序包,如 DATAIO 公司的 GENERICS 库、DATAIO 库等,以及 Synopsys 公司的一些库。

二、库的用法

在 VHDL 语言中,库的说明语句总是放在实体单元前面。这样在设计实体内的语句就可以使用库中的数据和文件。由此可见,库的用处在于使设计者可以共享已经编译过的设计成果。VHDL 允许在一个设计实体中同时打开多个不同的库,但库之间必须是相互独立的。

USE 语句的使用将使所说明的程序包对本设计实体部分或全部开放,即是可视的。USE 语句的使用有两种常用格式:

USE 库名.程序包名.项目名;

USE 库名.程序包名.ALL;

第一语句格式的作用是向本设计实体开放指定库中的特定程序包内所选定的项目。

第二语句格式的作用是向本设计实体开放指定库中的特定程序包内所有的内容。

合法的 USE 语句的使用方法是将在 USE 语句说明中所要开放的设计实体对象紧跟在 USE 语句之后,例如语句:

```
USE IEEE.STD_LOGIC_1164.ALL;
```

表明打开 IEEE 库中的 STD_LOGIC_1164 程序包,并使程序包中所有的公共资源对于本语句后面的 VHDL 设计实体程序全部开放,即该语句后的程序可任意使用程序包中的公共资源。这里用到了关键词 ALL 代表程序包中所有资源。

```
LIBRARY IEEE ;
USE IEEE.STD_LOGIC_1164.STD_ULOGIC ;
USE IEEE.STD_LOGIC_1164.RISING_EDGE ;
```

4.4.9 程序包 PACKAGE

已在设计实体中定义的数据类型、子程序或数据对象对于其他设计实体是不可用的,或者说不可见的。为了使已定义的常数、数据类型、元件调用说明以及子程序能被更多的 VHDL 设计实体方便地访问和共享,可以将它们收集在一个 VHDL 程序包中。多个程序包可以并入一个 VHDL 库中,使之适用于更一般的访问和调用范围。这一点对于大系统开发、多个或多组开发人员同步并行工作显得尤为重要。

通常程序包中的内容应具有更大的适用面和良好的独立性,以供各种不同设计需求的调用,如 STD_LOGIC_1164 程序包定义的数据类型 STD_LOGIC 和 STD_LOGIC_VECTOR。一旦定义了一个程序包,各种独立的设计就能方便地调用。

程序包的一般语句结构如下:

程序包说明部分:

```
PACKAGE 程序包名 IS
```

```
    程序包首说明部分
```

```
END 程序包名;
```

程序包包体部分:

```
PACKAGE BODY 程序包名 IS
```

```
    程序包包体说明部分以及包体内
```

```
END 程序包名;
```

程序包的结构由程序包的说明部分即程序包首和程序包的内容部分即程序包体两部分组成。一个完整的程序包中,程序包首的程序包名与程序包体的程序包名是同一个名字。

一、程序包首

程序包首的说明部分可收集多个不同的 VHDL 设计所需的公共信息,其中包括数据类型说明、信号说明、子程序说明及元件说明等。所有这些信息虽然也可以在每一个设计实体中进行逐一单独的定义和说明,但如果将这些经常用到的、并具有一般性的说明定义放在程序包中供随时调用,显然可以提高设计的效率和程序的可读性。

二、程序包体

程序包体将包括在程序包首中已定义的子程序的子程序体。程序包体说明部分的组成内容可以是 USE 句(允许对其他程序包的调用)、子程序定义、子程序体、数据类型说明、子类型说明和常数说明等。对于没有具体子程序说明的程序包体可以省去。

常用的预定义的程序包如下。

1. STD_LOGIC_1164 程序包

STD_LOGIC_1164 程序包是 IEEE 库中最常用的程序包,是 IEEE 的标准程序包。其中包含了一些数据类型、子类型和函数的定义,这些定义将 VHDL 扩展为一个能描述多值逻辑(即除具有 0 和 1 以外还有其他的逻辑量,如高阻态“Z”、不定态“X”等)的硬件描述语言,很好地满足了实际数字系统的设计需求。STD_LOGIC_1164 程序包中用得最多和最广得是定义了满足工业标准的两个数据类型 STD_LOGIC 和 STD_LOGIC_VECTOR,它们非常适合于 FPGA/CPLD 器件中多值逻辑设计结构。

2. STD_LOGIC_ARITH 程序包

STD_LOGIC_ARITH 预先编译在 IEEE 库中,是 Synopsys 公司的程序包。此程序包在 STD_LOGIC_1164 程序包的基础上扩展了三个数据类型 UNSIGNED、SIGNED 和 SMALL_INT,并为其定义了相关的算术运算符和转换函数。

3. STD_LOGIC_UNSIGNED 和 STD_LOGIC_SIGNED 程序包

STD_LOGIC_UNSIGNED 和 STD_LOGIC_SIGNED 程序包都是 Synopsys 公司的程序包,都预先编译在 IEEE 库中。这些程序包重载了可用于 INTEGER 型及 STD_LOGIC 和 STD_LOGIC_VECTOR 型混合运算的运算符,并定义了一个由 STD_LOGIC_VECTOR 型到 INTEGER 型的转换函数。这两个程序包的区别是,STD_LOGIC_SIGNED 中定义的运算符考虑到了符号,是有符号数的运算。

程序包 STD_LOGIC_ARITH、STD_LOGIC_UNSIGNED 和 STD_LOGIC_SIGNED 虽然未成为 IEEE 标准,但已经成为事实上的工业标准,绝大多数的 VHDL 综合器和 VHDL 仿真器都支持它们。

4. STANDARD 和 TEXTIO 程序包

以上已经提到了 STANDARD 和 TEXTIO 程序包,它们都是 STD 库中的预编译程序包。

STANDARD 程序包中定义了许多基本的数据类型、子类型和函数。由于 STANDARD 程序包是 VHDL 标准程序包,实际应用中已隐性地打开了,所以不必再用 USE 语句另作声明。TEXTIO 程序包定义了支持文本文件操作的许多类型和子程序。在使用本程序包之前需加语句 USE STD.TEXTIO.ALL。

TEXTIO 程序包主要仅供仿真器使用。可以用文本编辑器建立一个数据文件,文件中包含仿真时需要的数据,然后仿真时用 TEXTIO 程序包中的子程序存取这些数据。在 VHDL 综合器中此程序包被忽略。

4.4.10 配置 CONFIGURATION

配置也是 VHDL 设计实体中的一个基本单元,在综合或仿真中,可以利用配置语句为确定整个设计提供许多有用信息。配置语句还能用于对元件的端口连接进行重新安排等。VHDL 综合器允许将配置规定对一个设计实体中的最高层设计单元,但只支持对最顶层的实体进行配置。但是通常情况下配置主要用在 VHDL 的行为仿真中。

配置语句的一般格式如下:

```
CONFIGURATION 配置名 OF 实体名 IS
    配置说明
END 配置名;
```

配置主要为顶层设计实体指定结构体,或为参与例化的元件实体指定所希望的结构体,以层次方式来对元件例化作结构配置。如前所述,每个实体可以拥有多个不同的结构体,而每个结构体的地位是相同的,在这种情况下,可以利用配置说明为这个实体指定一个结构体。

4.5 Quartus II 使用指南

4.5.1 Quartus II 开发工具的介绍

目前国内使用最多的可编程逻辑器件是由美国 ALTERA 公司和 Xilinx 公司生产的,他们各自为自己的芯片开发提供了高效集成的基于计算机的开发环境。ALTERA 公司提供 Quartus II 和 MAX+plus II 开发软件,而 Xilinx 公司提供 ISE(Integrated Software Environment)开发环境。这两种开发软件均支持 VHDL 语言和 Verilog HDL 语言以及 ABEL 语言的编程,但是 Quartus II 开发环境仅仅支持 ALTERA 公司的所有系列的可编程逻辑器件产品,而 MAX+plus II 开发软件是 ALTERA 公司早期的软件产品,目前几乎被淘汰了,并且 ALTERA 公司自己也要求用户改用 Quartus II 开发软件。基于 Quartus II 的日益广泛的应用范围,下面介绍如何应用 Quartus II 开发软件进行简单的 FPGA 设计——VHDL 程序输入、管脚分配、程序下载。

4.5.2 文本设计输入

Quartus II 开发工具有多种设计输入方式,这里只选择文本输入方式进行介绍。

文本设计输入方式是采用 Quartus II 开发工具中的文本编辑器来进行的,它可以接受 AHDL、VHDL 和 Verilog HDL 等多种硬件描述语言。一般而言,在系统速率较低、时间特性要求不是十分严格的情况下,常采用这种输入方式。下面以 VHDL 的文本输入为例对文本设计输入方式进行介绍。

一、建立工程项目

在采用 Quartus II 开发工具进行一个具体的设计之前,需要建立一个工程项目。本书采用 Quartus II v9.1 版本进行设计,在开发工具的主管理窗口中,建立工程项目的操作步骤如下。

执行菜单命令【File】→【New Project Wizard】,为自己的设计建立一个新项目,并指定目标器件或器件系列,如图 4.5.1 所示。

打开 Wizard 后,将出现建立新项目的介绍对话框,介绍在新建项目过程中所要完成的具体工作,如图 4.5.2 所示。

单击 Next 按钮,进入工程路径和工程名称的设置对话框。如图 4.5.3 所示,在工程项目属性设置对话框中,第一栏是创建新工程文件的目录路径,默认选择的是 Quartus II 的安装路径,请重新指定工程的路径。第二栏是创建该项目的工程文件名,第三栏是创建该项目工程的顶层实体名,这里输入的顶层实体名必须与之后设计文件的顶层实体名相同,默认的顶层实体名与工程名相同,也可以根据需求输入不同的顶层实体名,建议取和工程内容相关的、一目了然的工程文件名。以设计一个 2 选 1 多路选择器为例,可以设置工程文件路径 E:\quartus\MUX21,设置工程文件名是 MUX21,设置顶层实体名是 MUX21,然后单击 Next 按钮,进行下一步的设置。

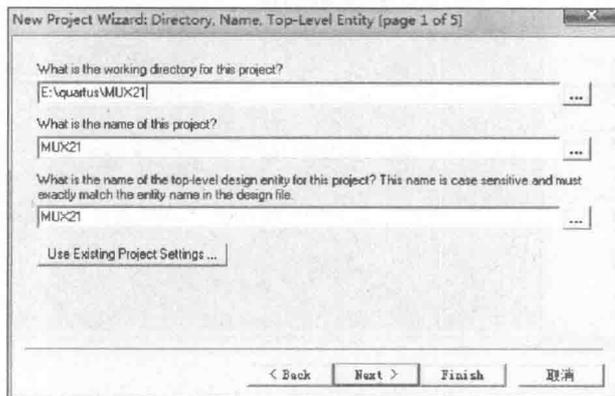


图 4.5.3 工程项目属性设置

Quartus II 软件的创建新工程项目向导共有 5 步。

第 1 步就是上面所讲的创建工程文件的目录路径、文件名、顶层实体名。

第 2 步是向创建的工程文件中添加已有文件和用户库,如果之前已经有设计好的文件,并且要在此工程用到这个文件(例如 VHDL 语言文件或原理图文件),那么可在此时将文件添加到工程中。如果没有完成的设计文件,点击 Next 进行下一步设置,如图 4.5.4 所示。

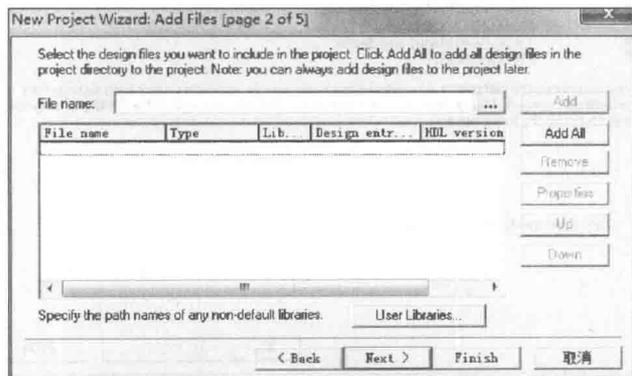


图 4.5.4 添加项目文件

第 3 步是为该工程选择可编程逻辑器件的器件族(这里选择 Cyclone III 器件族),以及选择一个具体的特定器件(这里选择 EP3C40F484C8 器件),如图 4.5.5 所示。

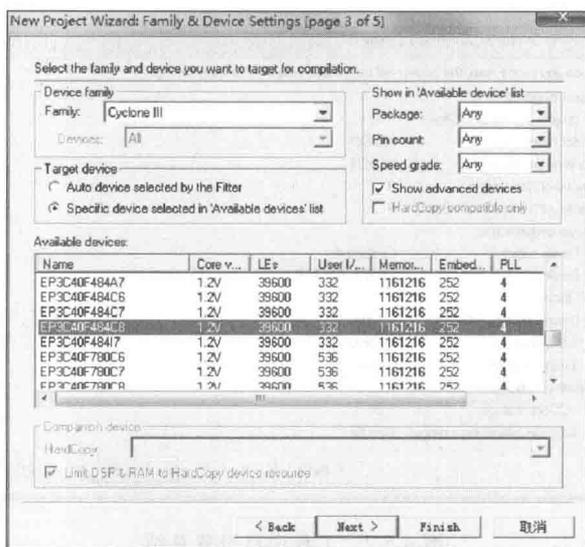


图 4.5.5 选择目标器件

第 4 步是对 EDA 工具的设置,在这里可以指定除 Quartus II 之外的用于设计输入、仿真、时序分析等的第三方 EDA 工具,如图 4.5.6 所示。

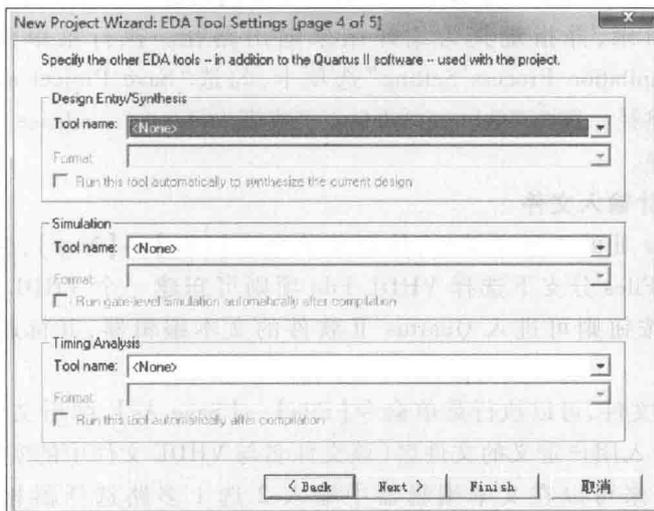


图 4.5.6 EDA 工具设置

第 5 步是对工程项目设置的总结,新建工程完成后,Quartus II 会自动总结用户的设置,在此确认设置是否正确,如图 4.5.7 所示。

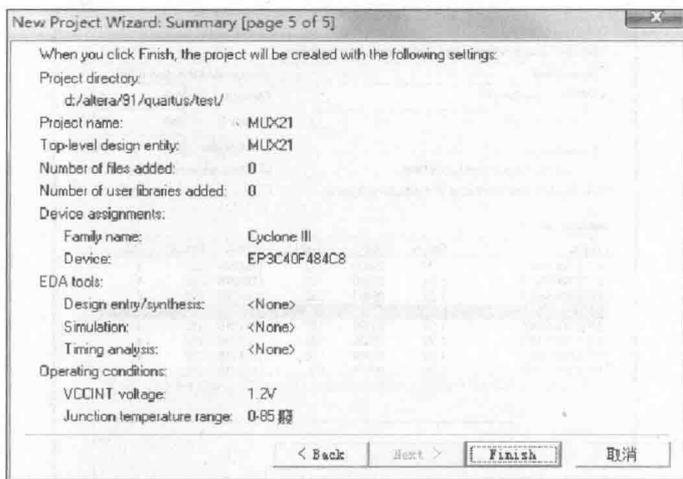


图 4.5.7 工程项目设置总结

最后单击 Finish 按钮完成新工程项目的创建,Quartus II 界面中“Project Navigator”的“Hierarchy”选项卡中会出现当前工程的工程名以及所选用的器件型号。

培养良好的文件布局习惯。Quartus II 默认把所有的编译结果文件统一放在工程根目录下,为了让 Quartus II 像 Visual Studio 等 IDE 一样有一个专门的路径存放编译结果文件,需要在工程根目录下新建一个目录,并指定其为编译结果输出路径。执行菜单命令【Assignments】→【Settings】,选中“Compilation Process Setting”选项卡,勾选“Save Project output files in specified directory”选项,输入路径一般为“debug(. \debug)”或者“release(. \release)”,如图 4.5.8 所示,单击“OK”按钮完成设置。

二、建立文本设计输入文件

在创建完 Quartus II 的工程文件后,执行菜单命令【File】→【New】,弹出新文件类型对话框,在打开的 Design Files 分支下选择 VHDL File 项则可新建一个 VHDL 文件(*.vhd),单击图 4.5.9 所示的 OK 按钮则可进入 Quartus II 软件的文本编辑器,其标题栏的默认文件名是 Vhdl1.vhd。

保存这个 VHDL 文件,可以执行菜单命令【File】→【Save As】,弹出文件保存对话框。在对话框的“File Name”输入用户定义的文件名(该文件名与 VHDL 文件中的实体名一致),在本例中输入“MUX21”。接下来可以在文本编辑器中输入 2 选 1 多路选择器相应的程序(参考【程序 4.1】,如图 4.5.10 所示)。

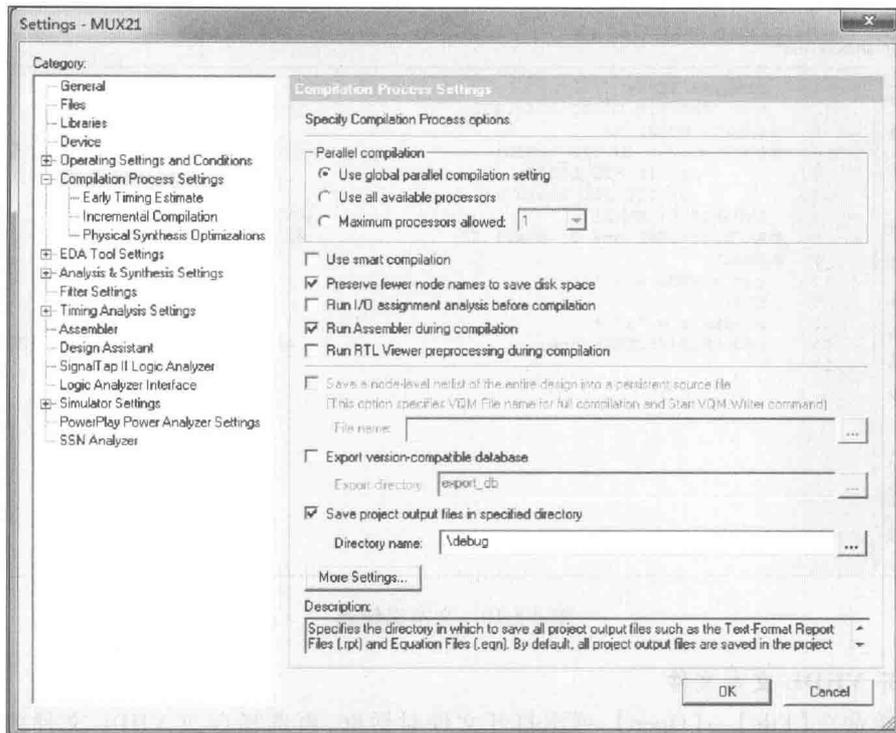


图 4.5.8 指定编译结果输出路径

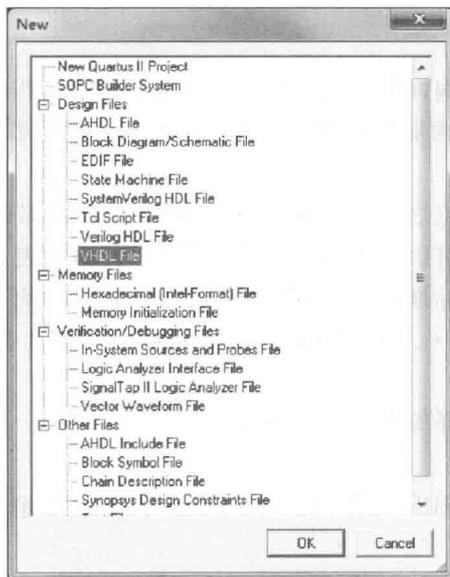


图 4.5.9 新建文件对话框

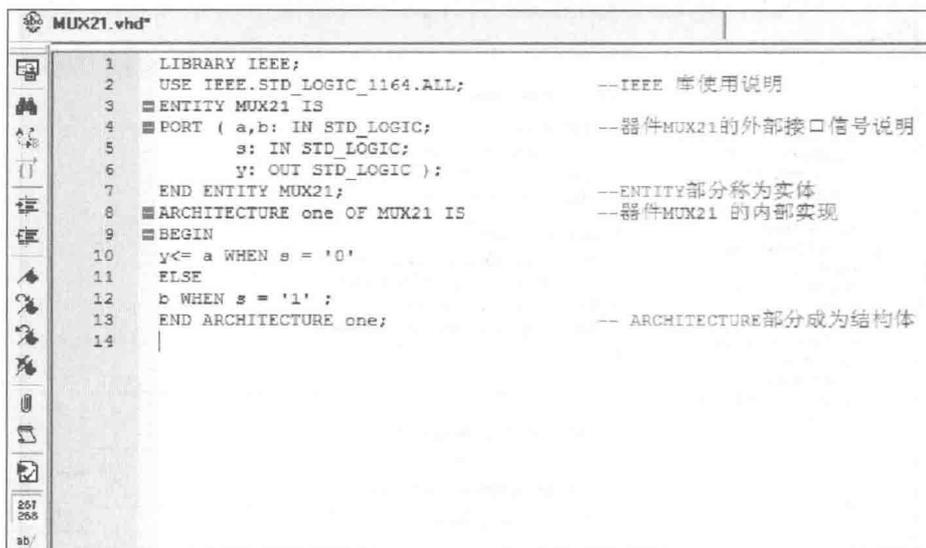


图 4.5.10 文本编辑器

三、打开 VHDL 文本文件

执行菜单命令【File】→【Open】，弹出打开文件对话框，再选择存放 VHDL 文件的文件路径；在“Files”中选择需要打开 VHDL 文件的文件名；点击 OK 按钮，完成操作。

4.5.3 设计文件编译

Quartus II 开发工具中的编译器(Compiler)可以用来检查工程项目中的错误，并进行逻辑综合，同时将项目的最终设计结构加载到 Altera 公司的可编程逻辑器件中去，并为设计校验和器件编译产生相应的输出文件。

Quartus II 开发工具的特点之一就是它可以将编译、综合两部分工作一起进行。

Quartus II 开发工具不但可以输出设计校验文件、仿真器网表文件(*.snf)、第三方 EDA 工具所用的网表文件(*.vo)和标准格式的 SDF 文件(*.sdo)等，而且可以输出可编程文件，包括用于编辑器下载的目标文件(*.pof)、用于在线可配置的 SRAM 目标文件(*.sof)和配置文件(*.sdf)等。

Quartus II 开发工具的编译综合分为如下步骤。

一、设置当前的项目文件

首先打开需要编译的 VHDL 文件，打开方式参考前文所述。这里我们打开 VHDL 文件 MUX21.vhd。

执行菜单命令【File】→【Open ...】，如图 4.5.11 所示。



图 4.5.11 打开文件

二、修改目标器件

设置了当前的项目文件后,设计者可以进行可编程逻辑器件的型号修改操作。执行菜单命令【Assignments】→【Device】,如图 4.5.12 所示,打开器件选择对话框,如图 4.5.13 所示。

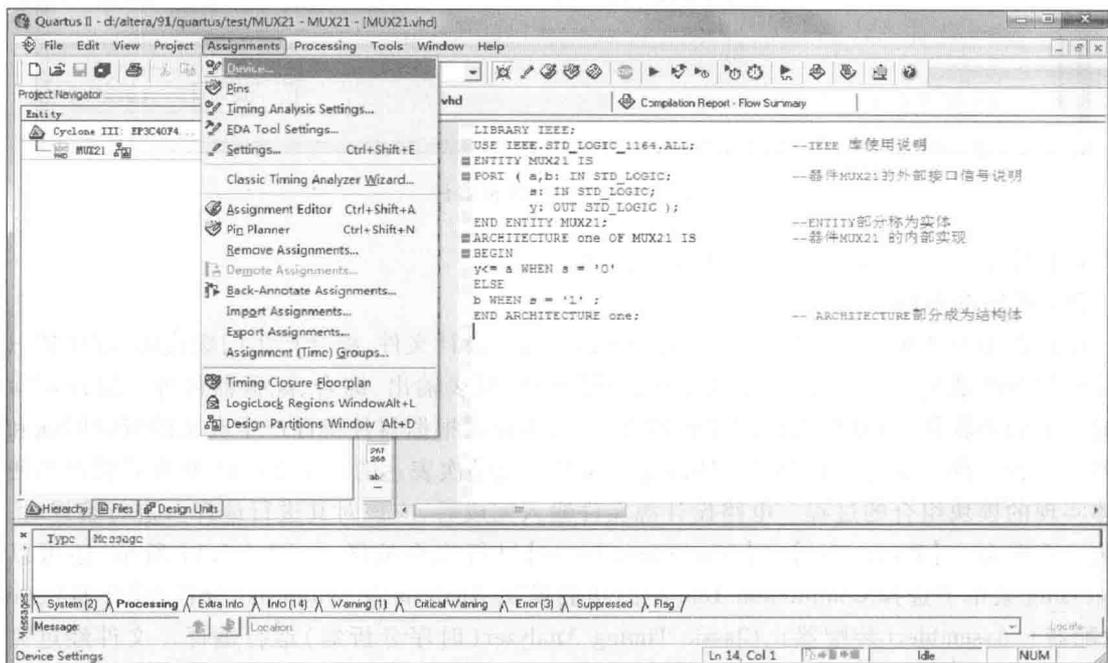


图 4.5.12 修改目标器件

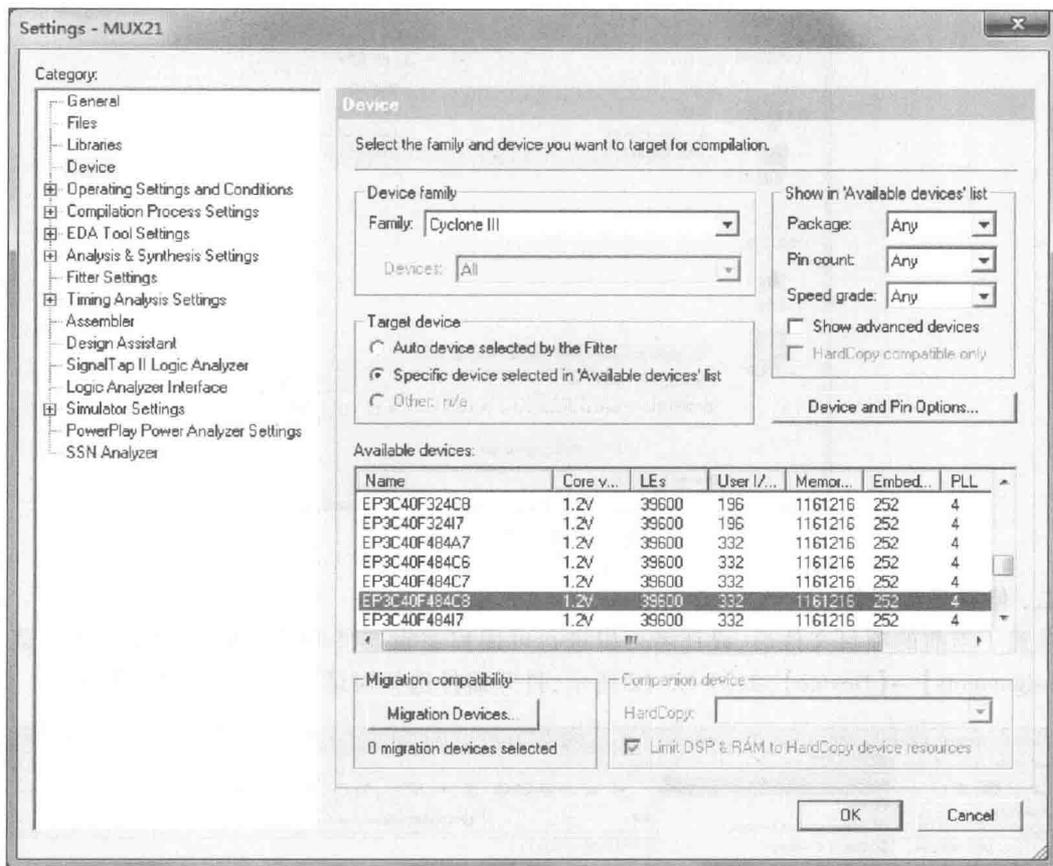


图 4.5.13 选择目标器件

选择完毕后,点击 OK 按钮,结束器件修改。

三、程序的编译

编译类型有两种:第一种是综合并输出网表,编译设计文件,综合产生门级代码,编译器只产生估算的延时数值;第二种是完全的编译,包括编译、网表输出、综合、配置器件等。编译器除了完成以上的步骤外,还要将设计配置到器件中去,编译器根据器件特性产生真正的延时时间和给器件的配置文件。综合(Synthesis)是将用行为和功能层次表达的电子系统转换为低层次的便于具体实现的模块组合的过程。电路设计源文件输入完成后,应该对其进行编译。进行编译时,可以执行菜单命令【Processing】→【Start Compilation】进行完全编译,如图 4.5.14 所示;还可以在 Processing 菜单下选择 Compilation Tool 中的功能模块 Analysis & Synthesis(分析/综合器)、Fitter(适配器)、Assembler(装配器)、Classic Timing Analyzer(时序分析器)运行编译。文件经过完全编译后,会产生编译报告以及相应的错误和警告信息。如果有错误,需对程序进行修改,直到出现编译成功提示信息,如图 4.5.15 所示。

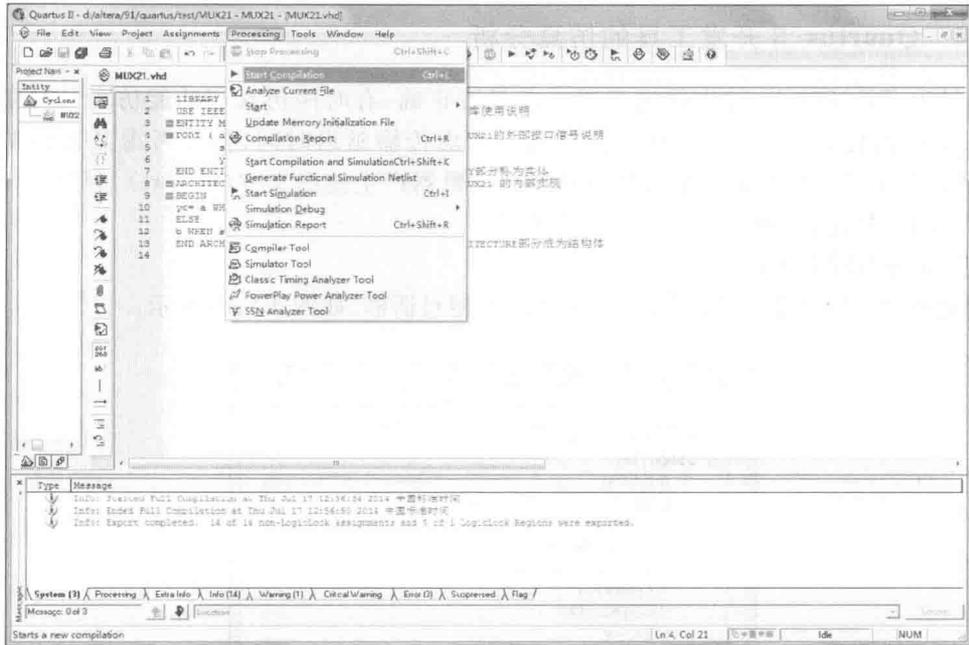


图 4.5.14 程序编译

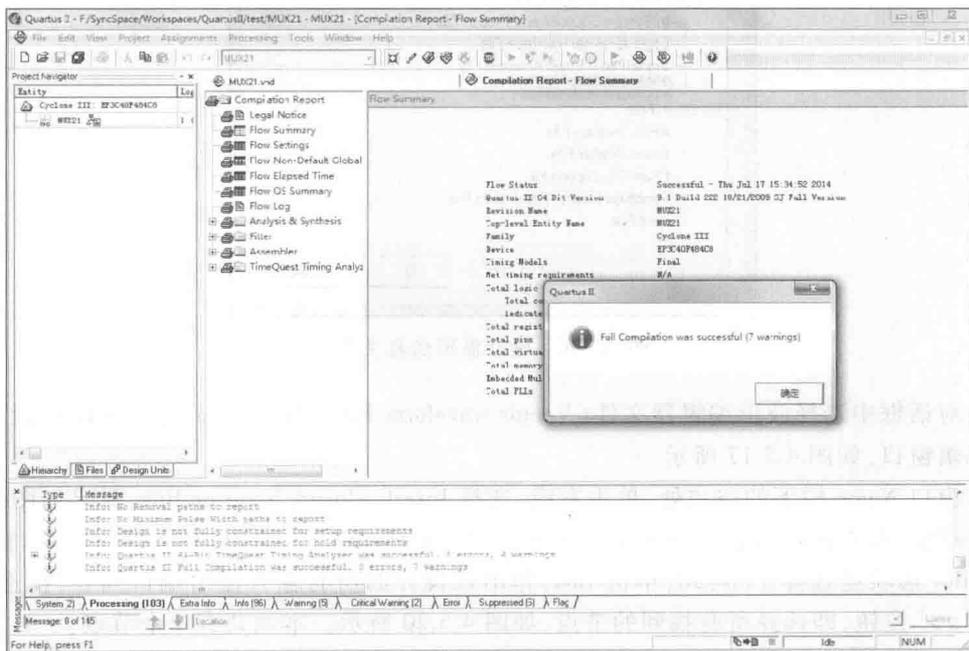


图 4.5.15 编译报告

4.5.4 Quartus II 开发工具的仿真分析

仿真是为了验证输入-输出间的逻辑功能是否正确,有时序仿真和功能仿真。时序仿真计算了电路运行的传输延迟时间,而功能仿真不考虑传输延迟时间,仅仅考虑其逻辑功能。在 Quartus II 开发工具的仿真输入信号是通过波形编辑器产生波形文件(*.vwf),通过*.vwf 文件和编译生产的文件连接,就可以实现程序的仿真。

一、建立仿真波形文件

执行菜单命令【File】→【New】,弹出新文件类型对话框,如图 4.5.16 所示。

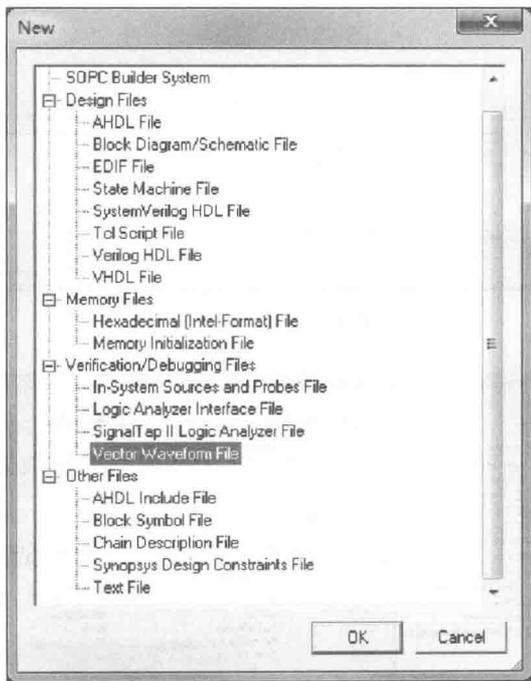


图 4.5.16 建立波形仿真文件

在该对话框中选择波形编辑器文件(Vector waveform File)类型,然后单击 OK 按钮,将会打开波形编辑窗口,如图 4.5.17 所示。

在该窗口 Name 栏下的空白处,单击右键,选择 Insert→Insert Node or Bus,弹出如图 4.5.18 所示插入节点窗口,在该窗口右边单击第三个按钮“Node Finder...”,弹出如图 4.5.19 所示窗口。

在 Filer 选项里选择 Pins: all,单击 List,弹出该设计的所有输入输出端口列表,在选择节点处,单击“>>”按钮,即选择所有找到的节点,如图 4.5.20 所示。本例共有 4 个节点。

根据需要,可以选择全部端口,也可以选择部分端口。单击 OK 返回插入节点窗口,单击 OK,弹出添加端口的项目波形文件编辑窗口,如图 4.5.21 所示。

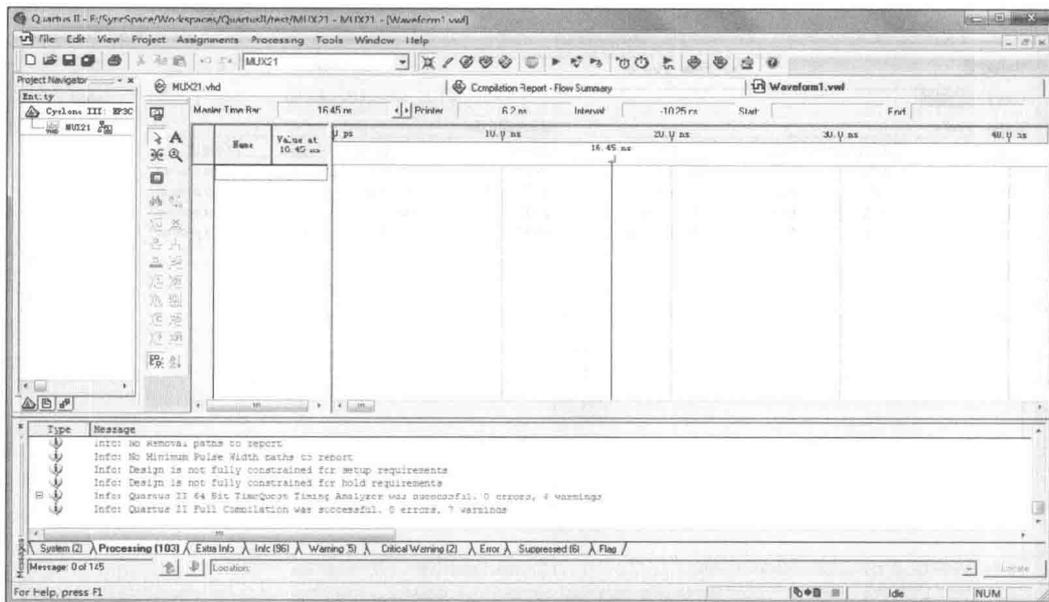


图 4.5.17 波形编辑窗口

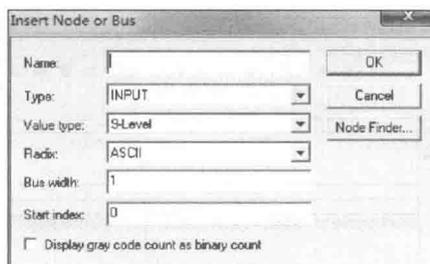


图 4.5.18 插入节点窗口

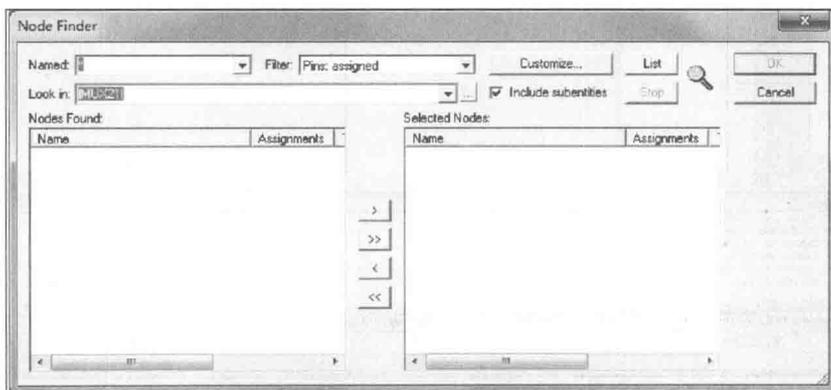


图 4.5.19 寻找节点窗口

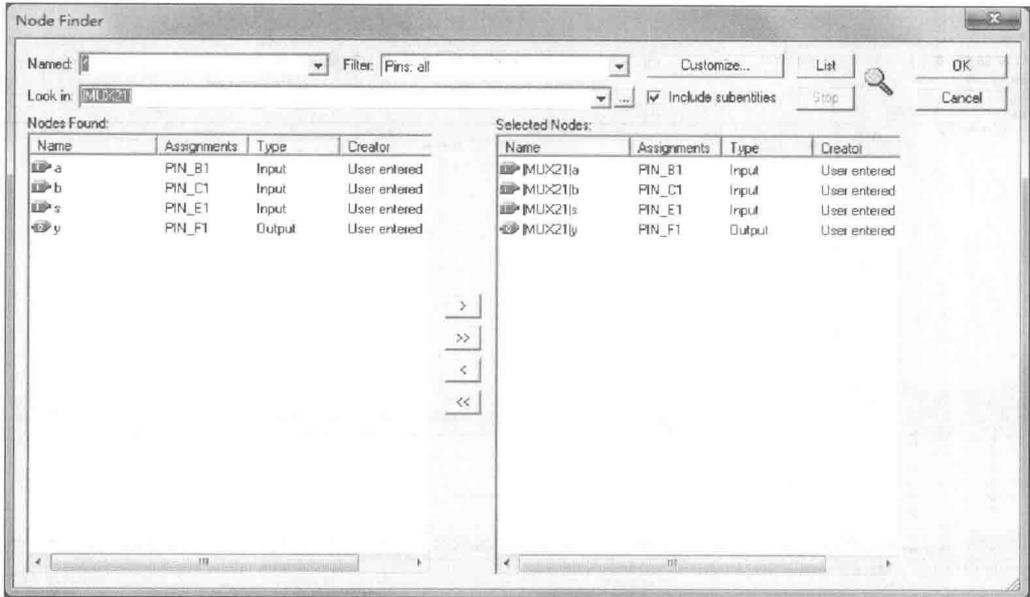


图 4.5.20 端口选择

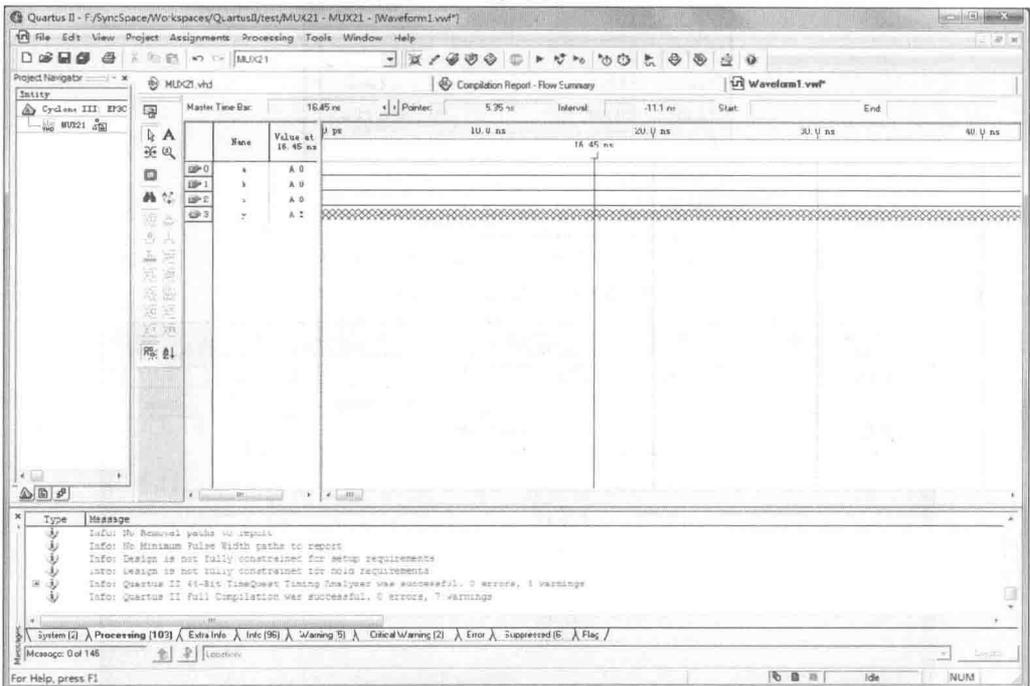


图 4.5.21 添加端口的波形编辑窗口

在实现仿真之前,需要进行一些必要的设定,如设定仿真时间长度、设定网格等。执行菜单命令【Edit】→【End Time】,如图 4.5.22,设定合适的仿真时间长度。执行菜单命令【Edit】→【Grid Size】,设定网格大小,如图 4.5.23 所示。

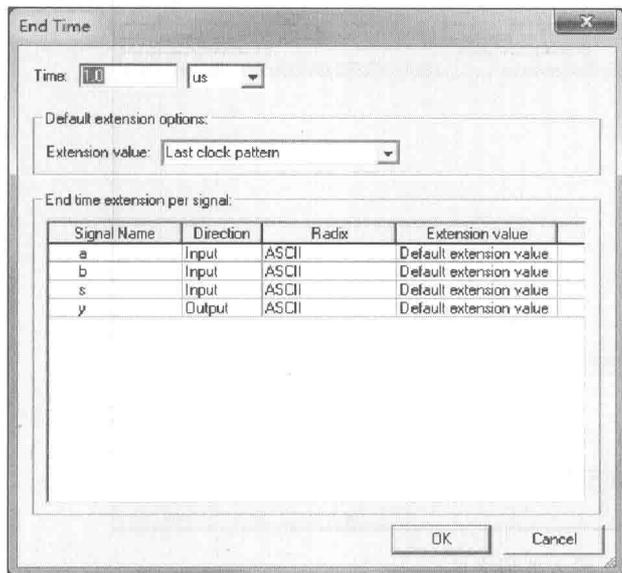


图 4.5.22 仿真时间长度设定

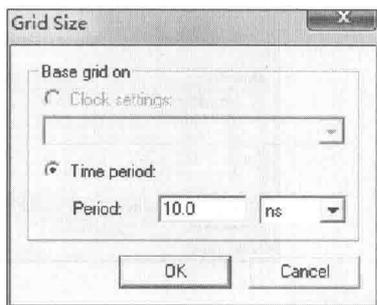


图 4.5.23 网格大小设定

二、编辑输入波形

单击某个端口,利用波形编辑工具对输入波形进行编辑,对于单个信号可以设定输入信号的值为高、低、高阻、弱高、弱低以及随机值等。编辑输入波形文件如图 4.5.24 所示。对仿真输入波形文件进行保存,保存文件名与工程名一致,保存文件形式为 vwf。本例中保存为 MUX21.vwf,保存路径为工程文件夹。

三、对设计文件进行仿真

第一次需要执行菜单命令【Processing】→【Simulator Tool】,在该窗口选择 Overwrite simulation input file with simulation results,选择该选项,可以把仿真结果进行保存。单击该窗口下的 Start 开始仿真,如图 4.5.25 所示,之后仿真时,可在菜单【Processing】下选择 Start Simulation 或者在软件主界面选择快捷方式即可。图 4.5.26 是时序仿真结果,从该结果可以看出,输出相对输入信号是存在延迟的,本例仿真正确,符合设计要求。

如果忽略延迟设计,只进行功能仿真,可在 Simulator Tool 窗口 Simulation mode 选项下选择 Functional,然后执行建立功能仿真网表 Generate Functional Simulation Netlist,如图 4.5.27 所示。单击 Start 按钮,开始逻辑功能仿真,仿真结果没有延迟和毛刺信号,如图 4.5.28 所示。

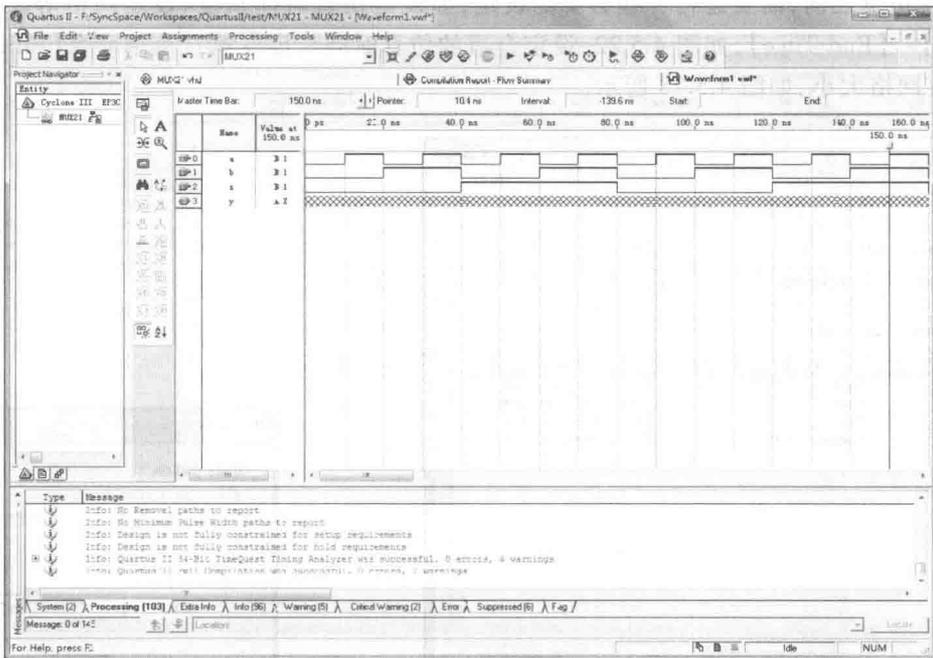


图 4.5.24 输入波形文件

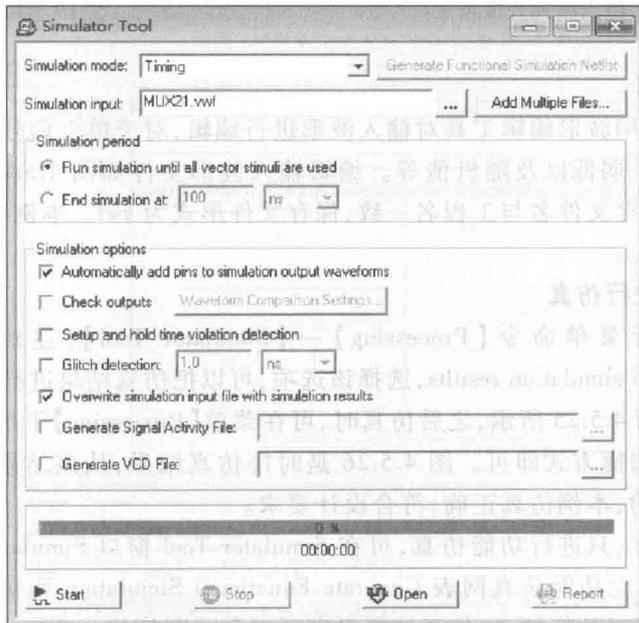


图 4.5.25 仿真工具设定窗口

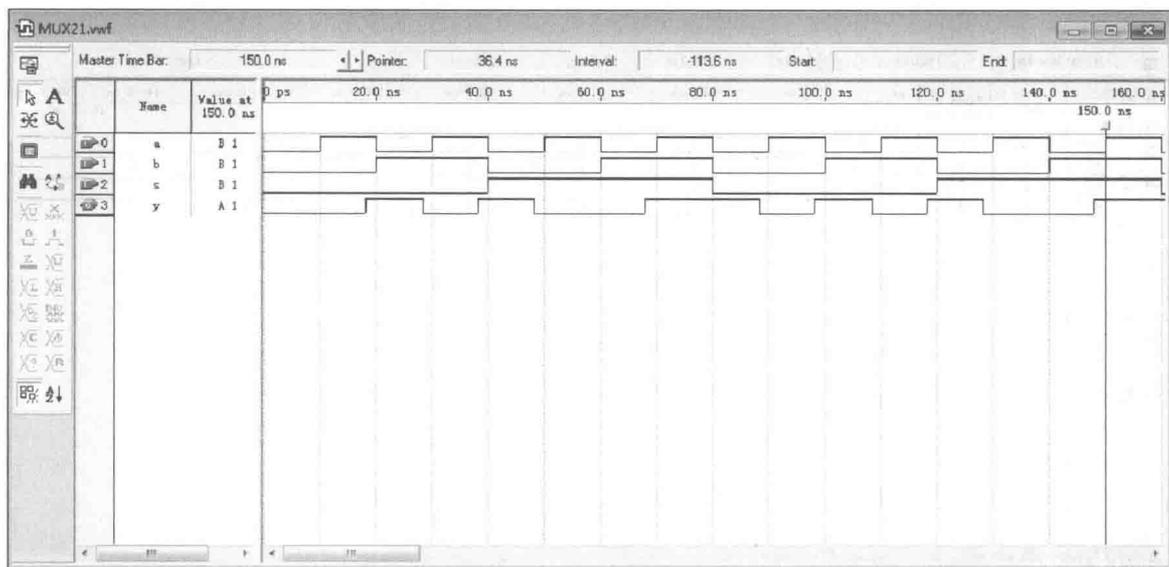


图 4.5.26 时序仿真结果

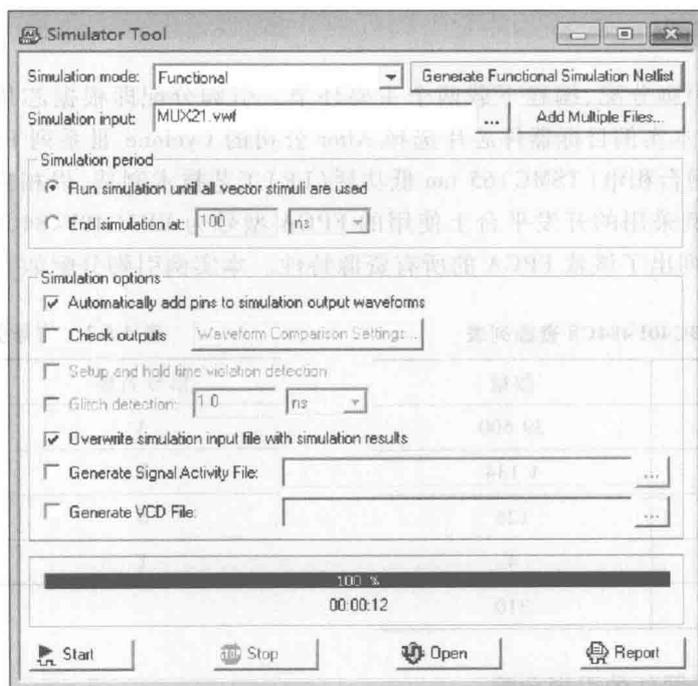


图 4.5.27 逻辑功能仿真设置

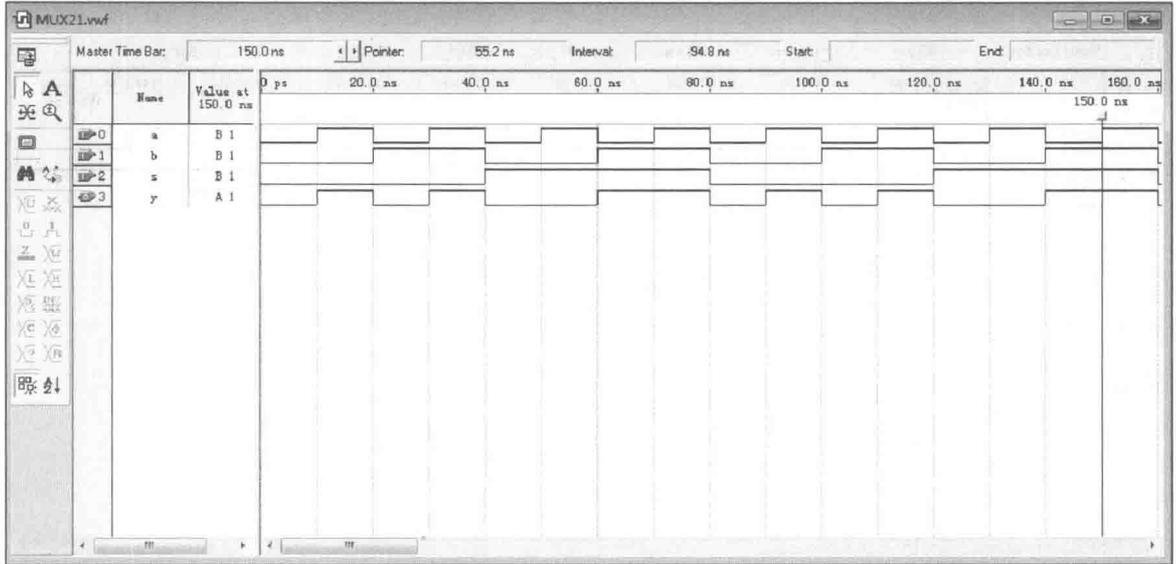


图 4.5.28 逻辑功能仿真结果

4.5.5 器件编程

器件编程包括引脚分配、编程下载两个主要环节。引脚分配即根据芯片资源,设计平台结构,进行引脚锁定。本实例目标器件芯片选择 Alter 公司的 Cyclone III 系列 FPGA,该系列 FPGA 是 2007 年推出,采用台积电(TSMC)65 nm 低功耗(LP)工艺技术制造,以相当于 ASIC 的价格实现了低功耗。本实例采用的开发平台上使用的 FPGA 型号为 EP3C40F484C8,采用 484 引脚的 BGA 封装,表 4.5.1 列出了该款 FPGA 的所有资源特性。本实例引脚分配如表 4.5.2 所示。

表 4.5.1 EP3C40F484C8 资源列表

资源	数量
Les	39 600
RAM(KBits)	1 134
内嵌乘法器	126
PLLs	4
用户可用 I/O	310

表 4.5.2 信号引脚分配表

信号名称	引脚编号
A	B1
B	C1
S	E1
Y	F1

一、可编程逻辑器件的引脚分配

引脚分配是可编程逻辑器件开发过程中的一个重要步骤,设计者在文本编辑器中输入的 VHDL 设计文件,只是对器件内部相应的电路输入和输出信号名称做了定义,但是没有

给出输入和输出信号与可编程逻辑器件实际的具体引脚对应的连接。如果需要利用 FPGA 开发平台对已经设计好的电路进行验证,就要将设计好的电路的输入与输出信号分配到目标器件的引脚上。本实例中的输入信号有:a、b 和 s;输出信号有:y。该引脚设定工作的具体操作如下。

执行菜单命令【Assignments】→【Pins】,打开引脚设置对话框,如图 4.5.29 所示。

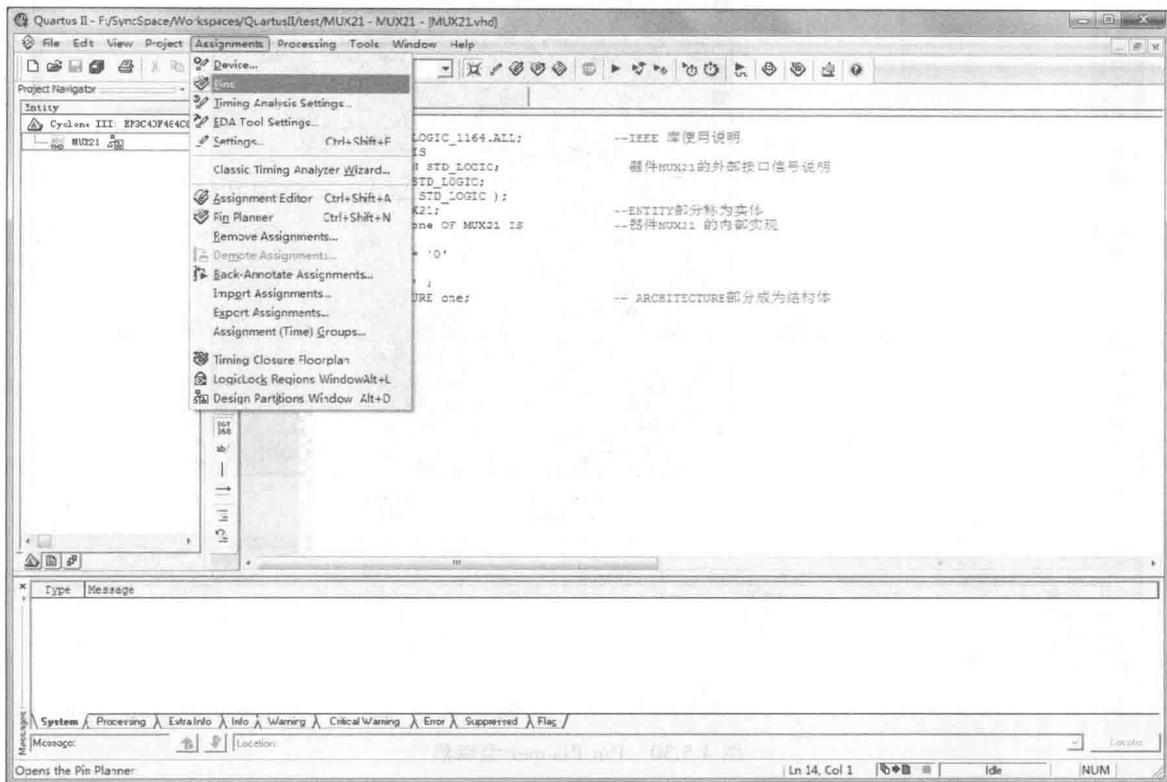


图 4.5.29 引脚设置

在打开的 Pin Planner 编辑器中进行引脚分配,如图 4.5.30 所示。在 Pin Planner 编辑器的下方,双击 location 栏目,为添加的端口进行引脚配置。具体引脚的配置如图 4.5.30 所示。

将未使用的管脚设置为三态。EP3C40F484C8 FPGA 共有 484 个管脚,本实验中,只用了其中的 4 个,还有大量的管脚在本实验中并未使用,Quartus II 默认这些未使用的管脚接地(逻辑 0),全部处于工作状态。这样 FPGA 工作时的耗电量将增加,长时间工作后 FPGA 会发烫,这样将大大降低 FPGA 的工作寿命。因此,需要对所有未使用的管脚进行设置。

双击 Project Navigator 的 Hierarchy 标签栏中的 Cyclone III:EP3C40F484C8,如图 4.5.31 所示,弹出如图 4.5.32 所示的界面。

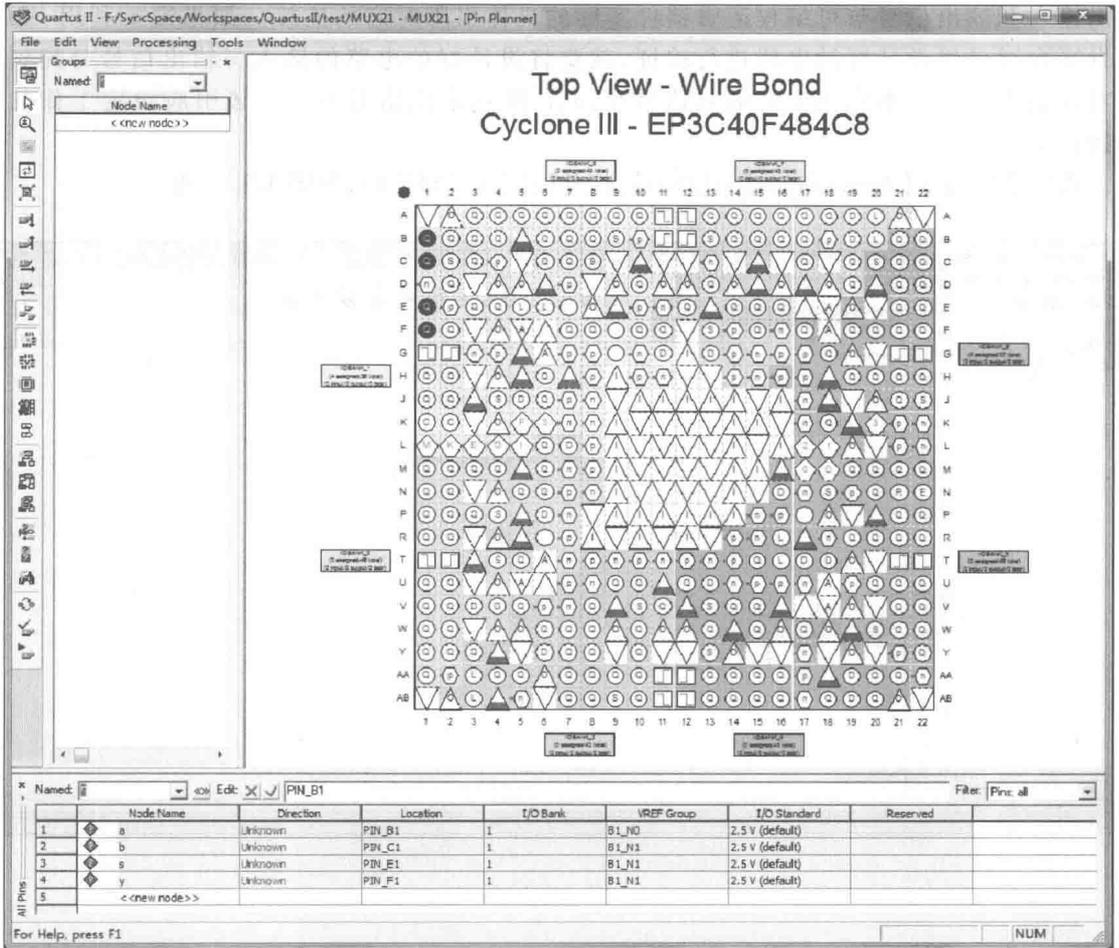


图 4.5.30 Pin Planner 编辑器



图 4.5.31 项目导航图

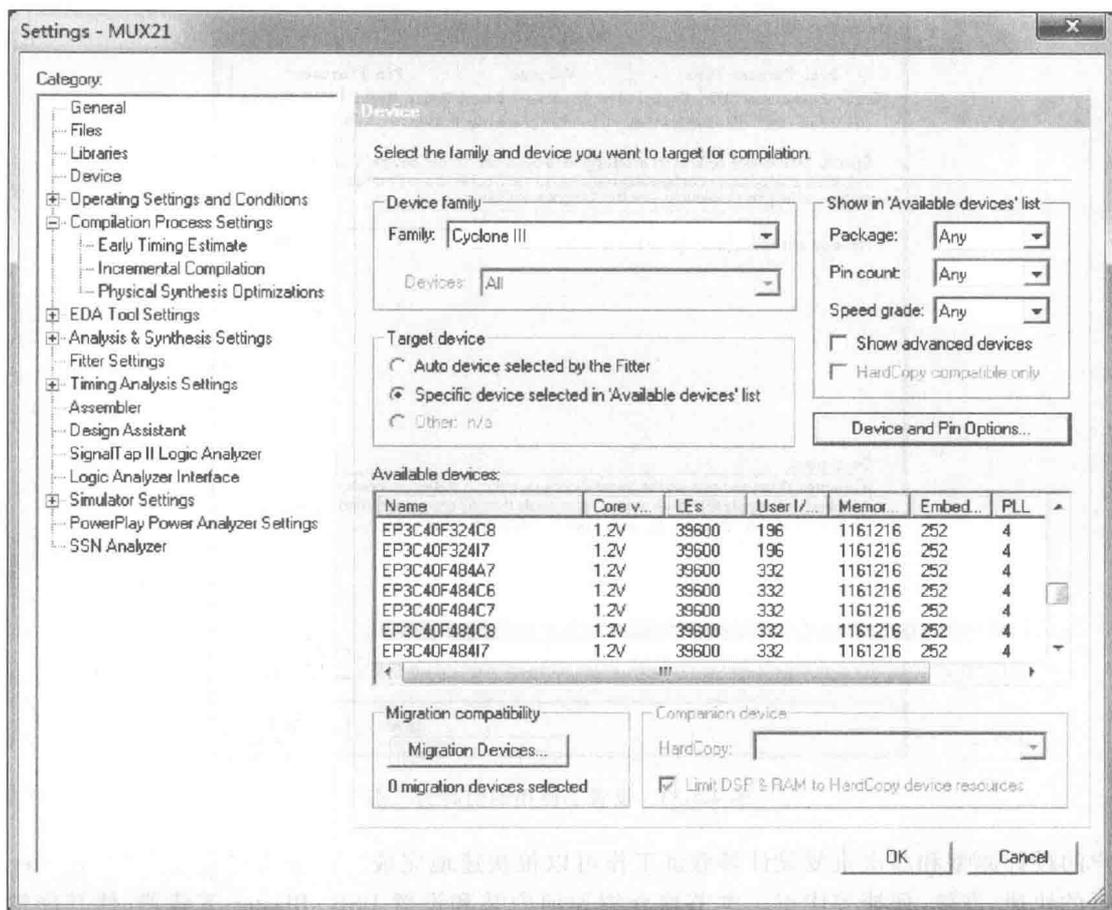


图 4.5.32 选择引脚设置

在图 4.5.32 中单击“Device and Pin Options”按钮,弹出如图 4.5.33 所示界面,打开“Unused Pins”选项卡,在“Reserve all unused pins”下拉框中选择“As input tri_stated”项,单击“OK”按钮,返回 Quartus II 界面。

二、设计项目的再编译

执行菜单命令【Processing】→【Start Compilation】,弹出编译器工作窗口,点击“Start Compilation”按钮,进行项目的再次编译综合。程序编译完成后,就可以连接电路板,进行程序的下调试。

三、USB 调试器的安装与使用

USB-Blaster 下载器可以通过 USB 端口把 PC 机和目标器件相连接。通过 USB-Blaster 下载器,PC 机可以将配置数据下载到目标器件中。由于设计变更等可以很容易地下载到目标器件,

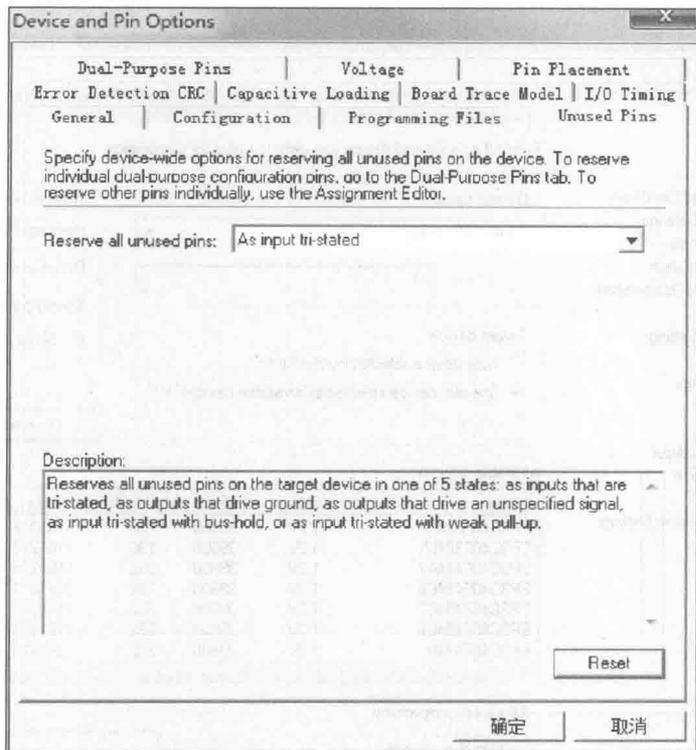


图 4.5.33 设置不使用的引脚为三态

用户的设计原型和多次重复设计等验证工作可以很快速地完成。这都要得益于 USB-Blaster 下载器的快速、高效、便捷等优点。本节将介绍如何安装和设置 USB-Blaster 下载器,使其能够正确地配置或编程器件。

按照如下指示,正确地连接 USB-Blaster 下载器到目标板。

- (1) 关闭目标板电源。
- (2) 将 USB-Blaster 下载器与目标板的 10 针插头相连接,如图 4.5.34 所示。
- (3) 将 USB-Blaster 下载器的 USB 端插入 PC 机的 USB 接口。
- (4) 重新给目标板上电。

如果是第一次在装有 Windows 的 PC 上使用 USB-Blaster 下载器,操作系统会弹出“发现新硬件”的安装向导,提示发现新的硬件,需要安装驱动,在安装完 Quartus II 后,驱动会出现在 \Quartus II 系统安装目录\drivers\usb-blaster 目录下。安装完成后进入“设备管理器”,查看硬件安装是否正确。正确安装 USB-Blaster 驱动后,会在“通用串行总线控制器”中出现“ALTERA USB-Blaster”的设备。如图 4.5.35 所示。

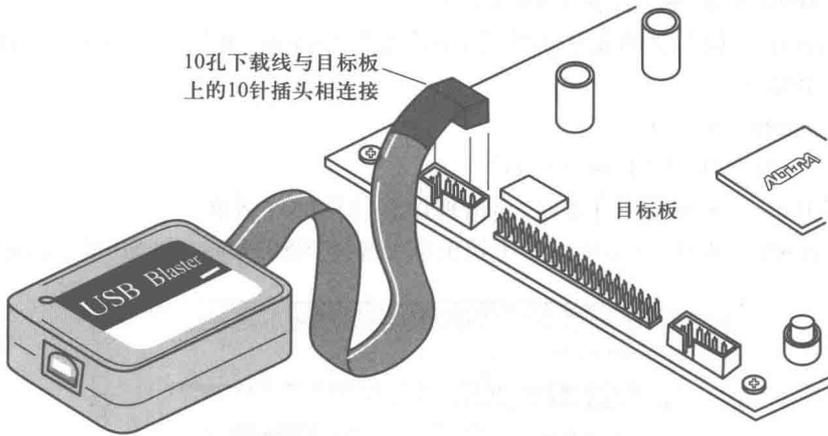


图 4.5.34 连接 USB-Blaster 下载线到目标板



图 4.5.35 查看安装的设备状况

四、在 Quartus II 软件中设置 USB-Blaster

在 USB-Blaster 下载器驱动安装正确后,还需要在 Quartus II 软件中进行适当的设置才能使用,具体设置步骤如下。

- (1) 启动 Quartus II 软件。
- (2) 选择“Tools”菜单下“Programmer”。
- (3) 点击【Hardware Setup…】按钮,会出现设置硬件的对话框。
- (4) 在“Currently selected hardware:”下拉菜单中选择“USB-Blaster[USB-0]”,如图4.5.36所示。

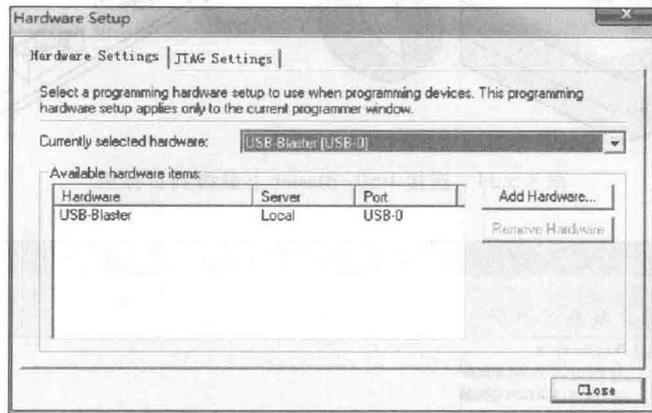


图 4.5.36 选择 USB-Blaster 为当前可用下载器

点击【Close】按钮,关闭硬件设置对话框。此时会返回到编程窗口,可以看到该窗口中列出了刚刚选择的下载电缆。

在编程窗口中的“Mode”下拉菜单中,选择需要的下载模式。表 4.5.3 列出了各种编程模式。USB-Blaster 下载电缆支持 Joint Test Active Group (JTAG)、Passive Serial Programming 和 Active Serial Programming 这三种下载模式。

表 4.5.3 编程模式列表

下载模式	模式描述
Joint Test Action Group (JTAG 模式)	编程或配置所有 Quautus II 软件中支持的除了 FLEX6000 以外的所有 Altera 器件
In-Socket Programming (套接字编程模式)	USB-Blaster 不支持该模式
Passive Serial Programming (被动串行模式)	编程或配置所有 Quautus II 软件中支持的除了 MAX3000 和 MAX7000 以外的所有 Altera 器件
Active Serial Programming (主动串行模式)	编程单片 EPCS1、EPCS4、EPCS16 和 EPCS64 等串行配置器件

五、目标文件加载

完成对器件的加载有两种形式：一种是对目标器件进行加载文件，即用 FPGA_JTAG 接口对 FPGA 芯片进行下载；一种是对目标器件的配置芯片进行加载，即用 FPGA_AS 接口对配置芯片进行下载。这里我们介绍对目标器件 EP3C40F484C8 进行加载的方法。

第一种加载方式方法如下。

- (1) 使用下载电缆将 PC 机与实验系统的 FPGA_JTAG 接口连接起来。
- (2) 选择 Quartus II v9.1 软件的 Tool>Programmer 命令或直接在工具栏点击按钮，进入编程器窗口，如图 4.5.37 所示。

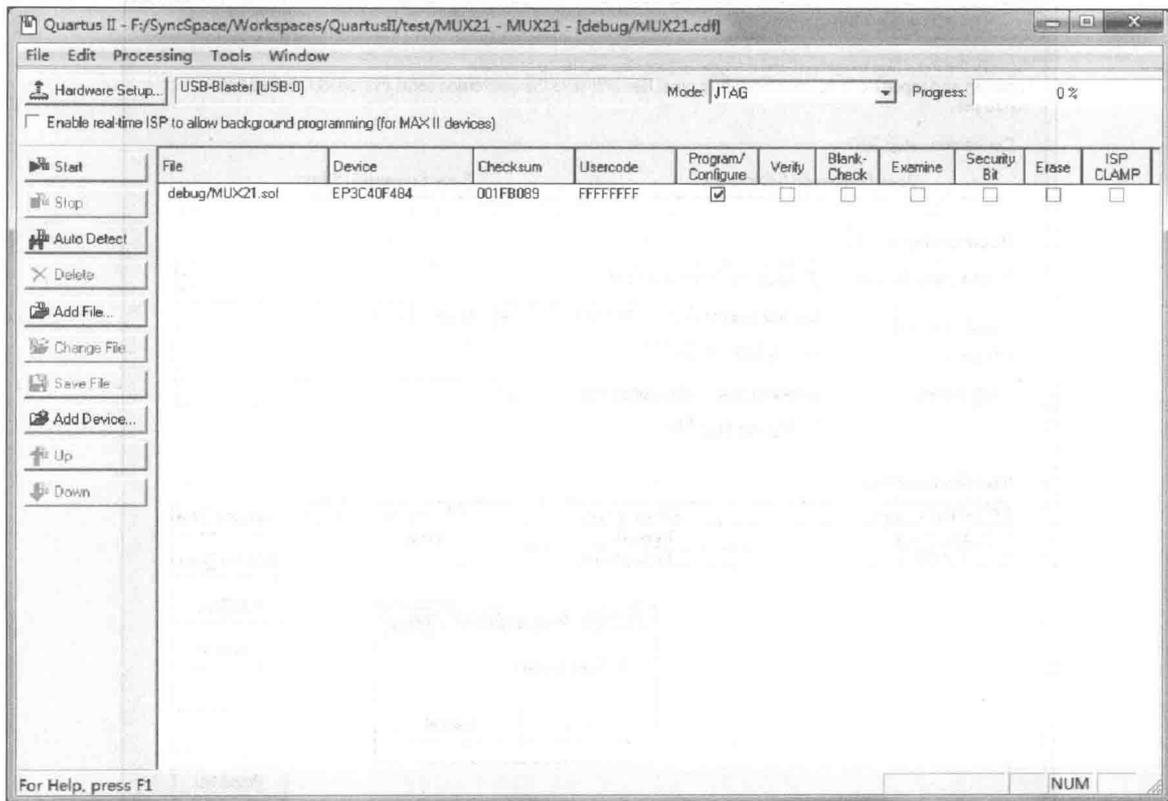


图 4.5.37 编辑器窗口

(3) 选择好加载文件（如果软件已运行一个工程，则在打开编程器的时候，编程器窗口会自动出现这个工程文件要加载到目标器件的文件，如果要加载其他文件可以从其他地方进行添加更改）后，点击 Program/Configure，在 Mode 栏中选择 JTAG 编程模式。再点击按钮，将文件下载到 FPGA 芯片 EP3C40F484C8 中，直到加载进度变为 100%，文件成功加载完成。

第二种加载方式方法如下。

(1) 使用下载电缆将 PC 机与实验系统的 FPGA_AS 接口连接起来。

(2) 选择 Quartus II v9.1 软件的 File>Convert Programming Files 命令, 出现生成文件对话框。在 Programming file type 栏后选择 Programmer Object File (.pof), 在 Configuration device 栏后选择 EPCS64, 将 File name 栏后名称改为 MUX21.pof。在 Input files to convert 栏中, 选中 SOF Data, 再点击右侧的 Add File... 将 MUX21.sof 文件添加到其中, 然后选中 MUX21.sof 文件, 再点击 Properties, 将 MUX21.sof 文件进行压缩以节省空间。设置方式如图 4.5.38 所示。点击 Generate 按钮, 即可生成适合从 FPGA_AS 接口下载的文件。

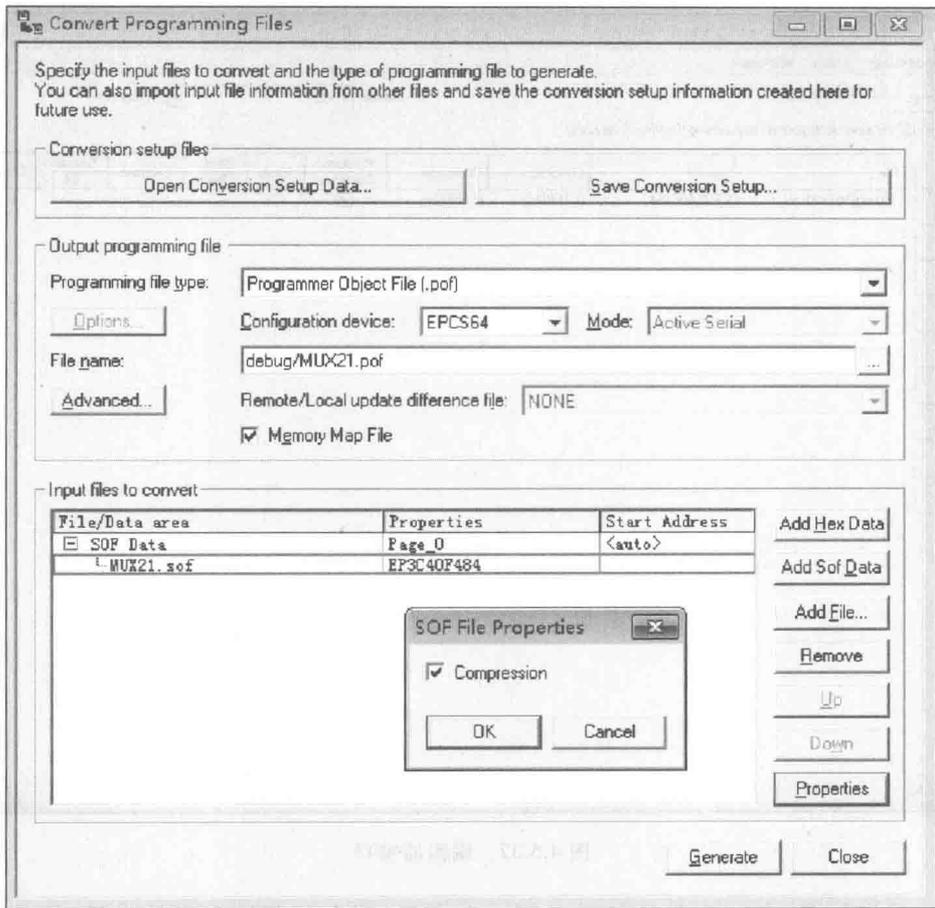


图 4.5.38 生成适合下载到配置芯片 EPCS64 文件对话框

(3) 选择 Quartus II v9.1 软件的 Tool>Programmer 命令或直接在工具栏点击按钮, 进入编程器窗口, 如图 4.5.39 所示。

(4) 在 Mode 栏后选择 Active Serial Programming 编程模式,选择好加载文件(如果软件已运行一个工程,则在打开编程器的时候,编程器窗口会自动出现这个工程文件要加载到目标器件的文件,如果要加载其他文件可以从其他地方进行添加更改)后,点选 Program/Configure,设置方式如图 4.5.39 所示。再点击 Start 按钮,将文件下载到配置芯片 EPCS64 中,直到加载进度变为 100%,文件成功加载完成。

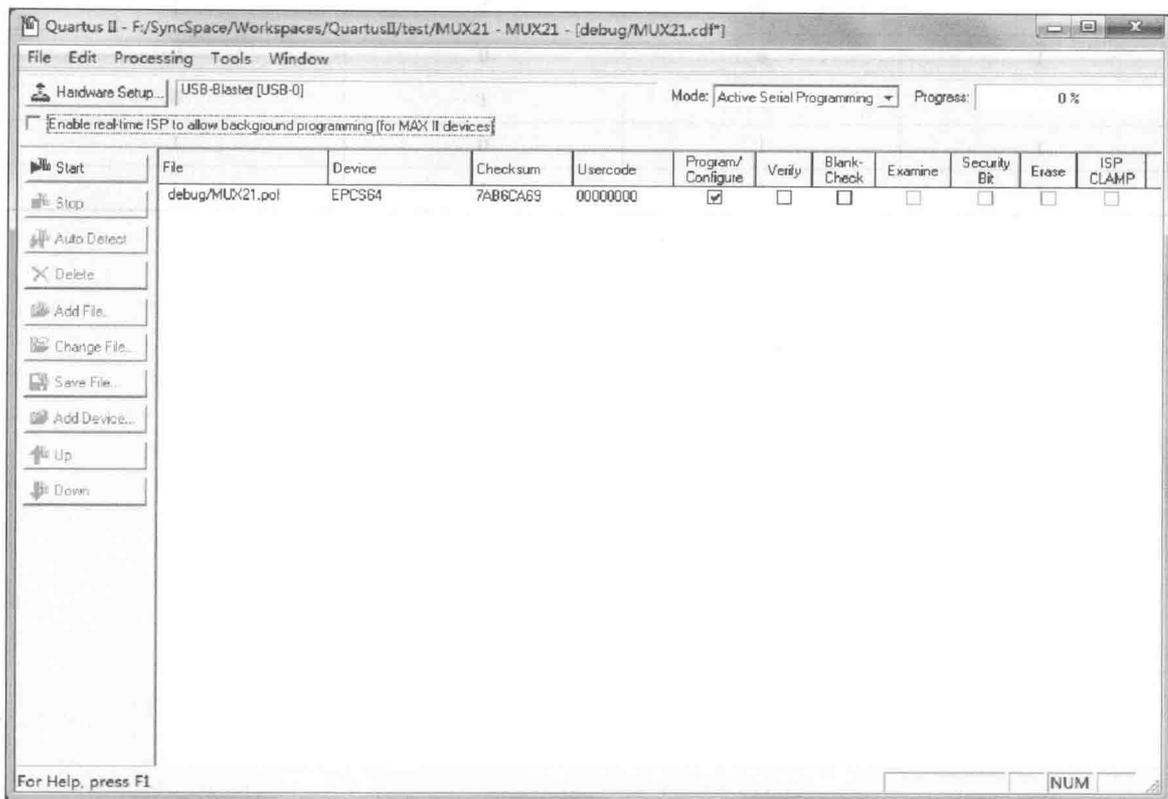


图 4.5.39 AS 编程模式加载对话框

目标文件成功加载到器件中之后,就可以对程序功能进行验证了,在信号与对应的引脚接入特定的输入信号,并观测输出信号的变化情况,验证其是否满足 2 选 1 数据选择器的功能真值表,如表 4.5.4 所示。

表 4.5.4 2 选 1 数据选择器的功能真值表

输入信号			输出信号
S	B	A	Y
0	0	0	0

续表

输入信号			输出信号
<i>S</i>	<i>B</i>	<i>A</i>	<i>Y</i>
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

第五章

电子技术综合实验 Quartus II 部分

题目一 3-8 译码器的设计

一、实验目的

- (1) 熟悉 VHDL 进程语句的设计。
- (2) 熟悉译码器的译码工作原理。
- (3) 熟悉 FPGA 开发板的下载流程。

二、实验内容

实现数字电路中的 3-8 译码器,包括三个数据输入端(D_0 、 D_1 、 D_2),三个使能控制信号(G_1 、 G_{2A} 、 G_{2B}),八个数据输出端(Y_0 、 Y_1 、 Y_2 、 Y_3 、 Y_4 、 Y_5 、 Y_6 、 Y_7)。 $Y_0 \sim Y_7$ 输出低电平有效。

三、实验原理

在数字电路中,3-8 译码器的电路如图 5.1.1 所示,真值表如表 5.1.1 所示。

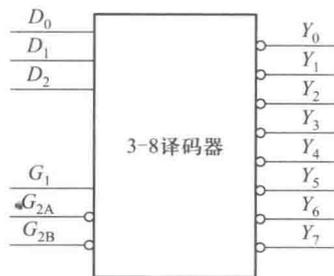


图 5.1.1 3-8 译码器电路图

表 5.1.1 3-8 译码器真值表

选通输入			二进制输入			译码输出							
G_1	G_{2A}	G_{2B}	D_0	D_1	D_2	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
×	1	×	×	×	×	1	1	1	1	1	1	1	1
×	×	1	×	×	×	1	1	1	1	1	1	1	1
0	×	×	×	×	×	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	0	1	1	0	1	1	1	1	1	1
1	0	0	0	1	0	1	1	0	1	1	1	1	1
1	0	0	0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	0	0	1	1	1	1	0	1	1	1
1	0	0	1	0	1	1	1	1	1	1	0	1	1
1	0	0	1	1	0	1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0

四、实验要求

- (1) 在 Quartus II 9.1 开发环境中, 根据要求设计出 3-8 译码器电路。
- (2) 用 Waveform 对门电路进行波形仿真, 验证结果的正确性。
- (3) 将程序下载到开发板, 用拨码开关做输入, LED 代替输出, 验证结果。

题目二 七段数码管显示

一、实验目的

- (1) 掌握七段数码管的显示原理。
- (2) 掌握 VHDL 实现 8421BCD 码在七段数码管上显示。

二、实验内容

用 VHDL 设计具有清零端、使能端、计数范围为 0~999 的计数器,并在七段数码管上显示 8421BCD 码的输出结果。

三、实验原理

1. 8421BCD 码

在数字系统中常用四位二进制代码来表示一位十进制数字 0、1、2、…、9,称之为二-十进制代码,即 BCD 码。将十进制数编成 BCD 码的电路,称为二-十进制(BCD)编码器。二-十进制编码的方案很多,若 BCD 编码器采用 8421 编码方案,称为 8421BCD 编码器。

2. 七段 LED 数码管

数字显示最常使用的器件就是七段数码管(如图 5.2.1),它的 7 个亮段组成了一个“8”,点亮不同亮段的组合就形成了数字 0~9。在数字钟、微波炉、电饭煲、洗衣机等电子产品中常常使用七段数码管来显示数字信息。



图 5.2.1 七段数码管

7 个亮段实际上就是 7 个条形的发光二极管。按顺时针方向,这 7 个亮段分别称为 a、b、c、d、e、f、g。七段数码管中亮段的发光原理和普通的发光二极管一致,可以把这 7 个亮段看成 7 个发光二极管。根据内部 7 个发光二极管的公共端不同,七段数码管有共阳(共阳极)和共阴(共阴极)两种,如图 5.2.2 所示。

如果要七段数码管显示数字“1”,只要点亮 b、c 两段即可;如要显示数字“5”,则需要点亮 a、f、g、c、d 段,如图 5.2.3 所示。

表 5.2.1 为共阴极七段数码管显示数值与输入值的对应表。

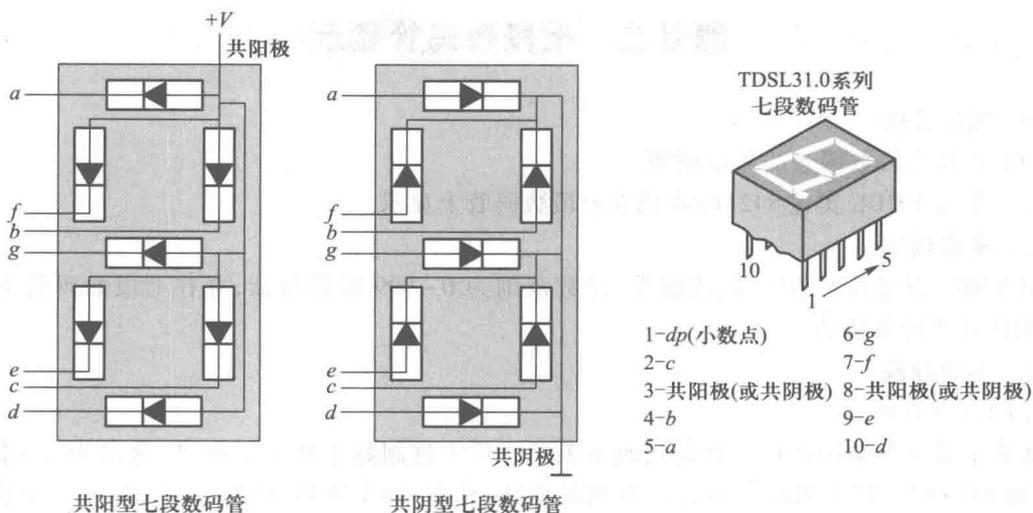


图 5.2.2 七段数码管内部结构

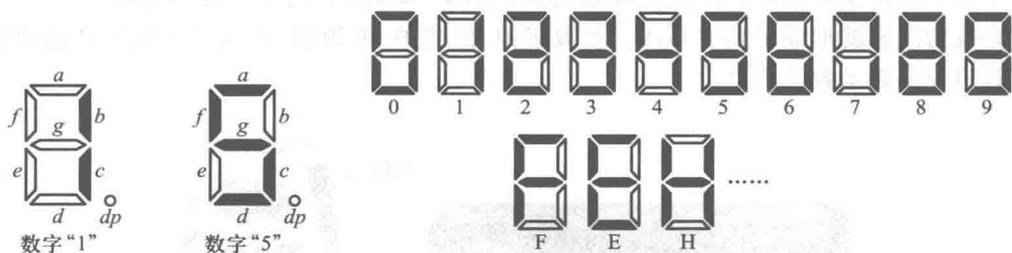


图 5.2.3 七段数码管显示原理

表 5.2.1 显示数值与输入对应表

输入 显示	a	b	c	d	e	f	g
0	1	1	1	1	1	1	0
1	0	1	1	0	0	0	0
2	1	1	0	1	1	0	1
3	1	1	1	1	0	0	1
4	0	1	1	0	0	1	1
5	1	0	1	1	0	1	1
6	1	0	1	1	1	1	1

续表

输入 显示	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>
7	1	1	1	0	0	0	0
8	1	1	1	1	1	1	1
9	1	1	1	1	0	1	1
A	1	1	1	0	1	1	1
B	0	0	1	1	1	1	1
C	1	0	0	1	1	1	0
D	0	1	1	1	1	0	1
E	1	0	0	1	1	1	1
F	1	0	0	0	1	1	1

四、实验要求

- (1) 在 Quartus II 9.1 开发环境中, 根据要求设计出 BCD 显示电路。
- (2) 将程序下载到开发板, 验证结果。

题目三 四位串行全加器的设计

一、实验目的

- (1) 掌握全加器的逻辑原理。
- (2) 掌握模块化程序设计方法。

二、实验内容

设计四位串行全加器。

三、实验原理

算术运算是数字运算的基本功能,更是计算机中不可缺少的组成单元。全加器能进行加数、被加数和低位来的进位进行相加,并根据求和结果给出该位的进位信号。

根据全加器的功能,列出的真值表如表 5.3.1 所示。全加器的逻辑图如图 5.3.1 所示。

表 5.3.1 全加器真值表

A_i	B_i	C_{i-1}	S_i	C_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

若有多位数相加,可采用并行相加串行进位的方式进行。四位的串行加法器如图 5.3.2 所示。

四、实验要求

- (1) 在 Quartus II 9.1 开发环境中,根据要求设计出四位串行加法器电路。
- (2) 将程序下载到开发板,以拨码开关做输入,LED 灯作输出,验证结果。

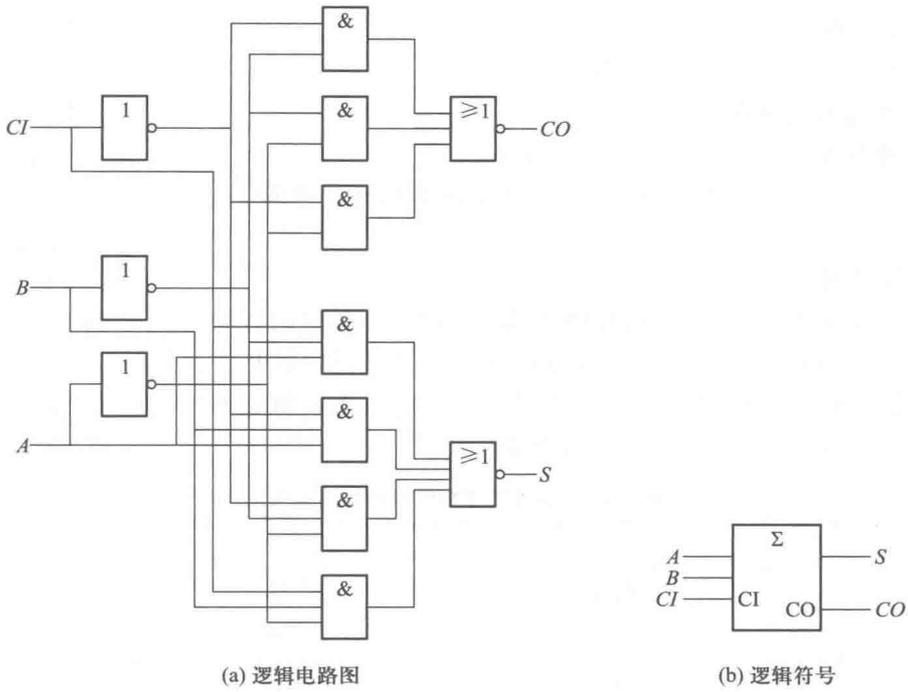


图 5.3.1 全加器的逻辑电路图和逻辑符号

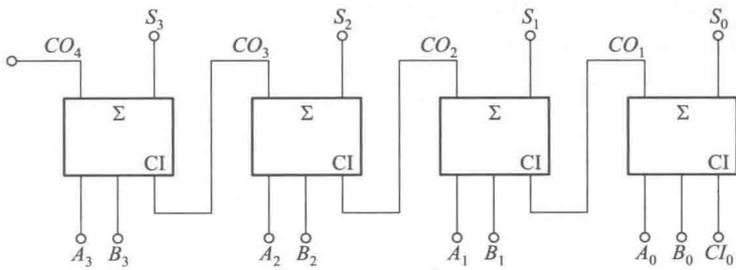


图 5.3.2 四位串行加法器示意图

题目四 六十进制计数器的设计

一、实验目的

- (1) 掌握同步计数器的设计方法。
- (2) 掌握分频电路设计方法。

二、实验内容

设计一个六十进制计数器,每 1 秒钟,计数值加 1,计数到 59 后清零,重新计数。

三、实验原理

同步计数器是在时钟脉冲(CLK)的控制下,构成计数器的各触发器状态同时发生变化的计数器。该计数器带有异步复位和计数允许功能端,四位二进制同步计数器的电路图如图 5.4.1 所示,真值表如表 5.4.1 所示。仿照四位二进制计数器的设计方法设计一个六十进制计数器。

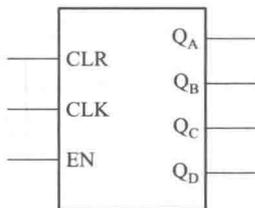


图 5.4.1 四位二进制同步计数器电路图

表 5.4.1 四位二进制同步计数器真值表

输入端			输出端			
CLR	EN	CLK	Q_A	Q_B	Q_C	Q_D
1	×	×	0	0	0	0
0	0	×	不变	不变	不变	不变
0	1	上升沿	计数值加 1			

四、实验要求

- (1) 在 Quartus II 9.1 开发环境中,根据要求设计出六十进制计数器电路。
- (2) 将程序下载到开发板,以 LED 灯显示结果。
- (3) 用七段数码管显示计数结果。

题目五 可编程彩灯控制器的设计

一、实验目的

- (1) 熟悉 VHDL 分频电路的设计。
- (2) 熟悉 FPGA 开发板的下载流程。

二、实验内容

设计一个可编程彩灯控制器,其中包括 6 种花型,且频率可调。

三、实验原理

(1) 设计一个可编程彩灯控制器,要求以某种节拍按一定规律改变彩灯的输入电流平均值,控制彩灯的亮与灭,按预定规律显示以下 6 种变换花型。

花型 1:8 路彩灯从左至右顺次渐亮,全亮后逆序渐灭。

花型 2:8 路彩灯从右至左顺次渐亮,全亮后逆序渐灭。

花型 3:8 路彩灯从中间到两边对称地逐次渐亮,全亮后仍由中间到两边逐次渐灭。

花型 4:8 路彩灯从两边到中间对称地逐次渐亮,全亮后仍由两边到中间逐次渐灭。

花型 5:8 路彩灯分两半,从左至右顺次渐亮,全亮后则灭。

花型 6:8 路彩灯分两半,从右至左顺次渐亮,全亮后则灭。

(2) 频率控制:设置按键 KEY 用来改变每种花型的循环频率(1 Hz、2 Hz、4 Hz、8 Hz)。

四、实验要求

- (1) 在 Quartus II 9.1 开发环境中,根据要求设计出彩灯控制器电路。
- (2) 将程序下载到开发板,用按键做频率控制,LED 做彩灯输出,验证结果。

题目六 乒乓球游戏机的设计与实现

一、实验目的

- (1) 掌握 VHDL 模块化设计电路方法。
- (2) 掌握数码管显示方法。

二、实验内容

设计一个乒乓球游戏机,LED 灯显示乒乓球运动路径,数码管显示得分。

三、实验原理

乒乓球比赛是由甲乙双方参赛,再加上裁判的三人游戏。乒乓球比赛游戏机是一种用发光二极管模拟乒乓球运动的电子游戏机,同时可以容纳三人游戏。

(1) 用 8 个以上的发光二极管排成一条直线或曲线,以中点为界,两边各代表参赛双方的位置。其中一只点亮的发光二极管指示球的当前位置。点亮的发光二极管依次从左到右或从右到左移动,其移动速度可以调节。

(2) 当球(点亮的发光二极管)运动到最后一位时,参赛者应能果断地按下位于自己一方的按钮开关,即表示启动球拍击球,若击中则使球向相反方向移动,若未击中,则对方得 1 分。

(3) 设置自动计分电路和胜局数统计电路。甲乙双方各用两位数码管进行计分显示,每计满 11 分为一局,并用计胜局电路统计双方的胜局数。

(4) 甲乙双方各设置一个发光二极管表示拥有发球权,每得 5 分自动交换发球权。

四、实验要求

- (1) 在 Quartus II 9.1 开发环境中,根据要求设计乒乓球游戏机。
- (2) 将程序下载到开发板,以按键做击球键,LED 灯代替球的路径,数码管显示分数,验证结果。

第六章

手工焊接技术简介

焊接是金属加工的基本方法之一,通常焊接技术分为熔焊、压焊和钎焊三大类。在电子产品制造过程中,使用最普遍、最有代表性的是锡焊方法。

6.1 锡焊原理

锡焊是将焊件和熔点比焊件低的焊料共同加热到锡焊温度,在焊件不熔化的情况下,焊料熔化并浸润焊接面,依靠二者原子的扩散形成焊件的连接。

一、合金层的生成

焊接的物理基础是“浸润”,浸润也叫“润湿”。液体在与固体的接触面上摊开,充分铺展接触,就叫做浸润。

锡焊的过程,就是通过加热,让铅锡焊料在焊接面上熔化、流动、浸润,使铅锡原子渗透到铜母材(导线、焊盘)的表面内,并在两者的接触面上形成 Cu_6Sn_5 的脆性合金层。

二、浸润与浸润角

在焊接过程中,焊料和母材接触所形成的夹角叫做浸润角,如图 6.1.1 所示。

三、锡焊必须具备的条件

1. 焊件必须具有良好的可焊性

可焊性是指在适当温度下,被焊金属材料与焊锡能形

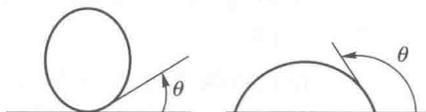


图 6.1.1 浸润角

成良好结合的合金的性能。

2. 焊件表面必须保持清洁
3. 要使用合适的助焊剂

助焊剂的作用是辅助热传导、去除氧化物、降低表面张力、防止再氧化。通常采用以松香为主的助焊剂。

4. 焊件要加热到适当的温度

适当的温度是使锡、铅原子获得足够的能量渗透到被焊金属表面的晶格中而形成合金。

焊接时间是指在焊接全过程中,进行物理和化学变化所需要的时间。一般 2~3 s,不超过 5 s。

6.2 焊接工具和材料

一、常用安装工具

电子产品装配过程都离不开常用安装工具,如图 6.2.1 所示,正确有效地使用安装工具能够提高产品组装的效率。



图 6.2.1 常用安装工具

1. 尖嘴钳

尖嘴钳的钳口长而细,钳口末端较小,钳口根部较粗。此种钳子用于折弯和加工细导线以及夹持小零件,不能用于弯折粗导线。

2. 斜口钳

斜口钳的钳口短,且有很平的刃口,其钳口位于侧面。这种斜口钳可用于剪细小导线,也可用于修整印刷电路板和装配中使用的塑料等。

3. 平口钳

平口钳也叫刻丝钳、电工钳,它具有厚型钳口,钳口结实带有纹路。主要用于重型作业。

4. 剥线钳

剥线钳的刃口有不同尺寸的槽形剪口,专用于剥去导线的绝缘皮。使用剥线钳时必须注意

把需要剥皮的导线放入合适的槽口,否则会损伤芯线。

5. 镊子

镊子有尖嘴镊子和圆嘴镊子两种。主要作夹具用具。焊接、拆卸小的电子元件时,用镊子作夹具,可使操作方便,有助于元器件散热。

6. 螺丝刀

螺丝刀有平口螺丝刀和十字螺丝刀,用于紧固螺丝钉,调整可调元件。调整元件或紧固螺丝钉时,所用螺丝刀型号一定要适当。过大,会因为调节力矩过大而损坏被调元件;过小,会因调节力矩不均而损坏螺丝刀。

二、电烙铁的种类

在电子产品组装和维修过程中常用的手工焊接工具是电烙铁。电烙铁具有许多品种和规格,按其加热方式来分,目前基本上有直热式和感应式两大类,并由此派生出许多不同的品种。常见的电烙铁有以下几种。

1. 外热式电烙铁

外热式电烙铁的规格很多,常用的有 25 W、45 W、75 W、100 W 等。电烙铁功率越大,烙铁头的温度越高。外热式电烙铁的结构如图 6.2.2 所示。它由烙铁头、烙铁芯、外壳、木柄、电源引线和电源插头等组成。由于发热的烙铁芯在烙铁头的外面,所以称为外热式电烙铁。

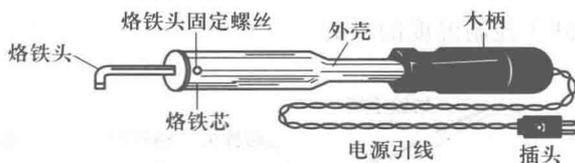


图 6.2.2 外热式电烙铁

2. 内热式电烙铁

内热式电烙铁的常用规格有 20 W、30 W、50 W 等几种。内热式电烙铁的烙铁芯是用比较细的镍铬电阻丝绕在瓷管上制成的,其电阻值约为 2.4 k Ω 左右(20 W),烙铁的温度一般可达 350 $^{\circ}\text{C}$ 左右。由于它的热效率高,内热式 20 W 电烙铁就相当于外热式 40 W 的电烙铁。由于内热式电烙铁有升温快、重量轻、耗电省、体积小、热效率高的特点,因而得到了普遍的应用。

3. 感应式电烙铁

感应式电烙铁也叫速热烙铁,俗称焊枪,如图 6.2.3 所示。它内部有一个变压器,这个变压器的二次侧实际只有一匝。所以,当其通电时,变压器的二次侧感应出大电流

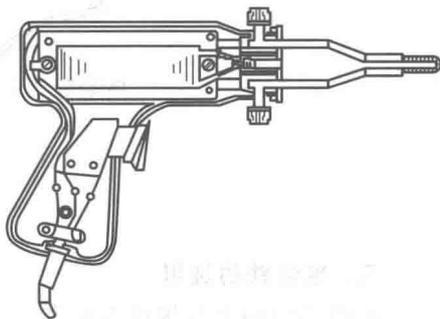


图 6.2.3 感应式电烙铁

通过加热体,使同它相连的烙铁头迅速达到焊接所需要的温度。由于这种烙铁加热速度快,一般通电几秒钟,即可达到焊接温度,因此不需要像直热式电烙铁那样持续加热。它的手柄上带有开关,特别适合于断续工作的使用。

4. 吸锡电烙铁

吸锡电烙铁是将活塞式吸锡器与电烙铁融为一体的拆焊工具。它具有使用方便、灵活、适用范围宽等特点。这种吸锡电烙铁的不足之处是每次只能对一个焊点进行拆焊。

吸锡电烙铁的使用方法是:接通电源预热(3~5分钟),然后将活塞柄推下并卡住,把吸锡铁的吸头前端对准欲拆焊的焊点,待焊锡熔化后,按下吸锡电烙铁手柄上的按钮,活塞便自动上升,将焊锡吸进气筒内。另外,吸锡器配有两个以上直径不同的吸头,可根据元器件引线的粗细进行选择。

5. 温控电烙铁

由于在焊接集成电路、晶体管元器件时,温度不能太高,焊接时间不能过长,否则就会因温度过高造成元器件的损坏,因而对电烙铁的温度要给以限制。而温控电烙铁就可以达到这一要求,如图6.2.4所示。由于温控电烙铁头内,装有带磁铁式的温度控制器,控制通电时间而实现温控,即给电烙铁通电时,烙铁的温度上升,当达到预定的温度时,因强磁体传感器达到了居里点而磁性消失,从而使磁芯触点断开,这时便停止向电烙铁供电;当温度低于强磁体传感器的居里点时,强磁体便恢复磁性,并吸动磁芯开关中的永久磁铁,使控制开关的触点接通,继续向电烙铁供电。如此循环往复,便达到了控制温度的目的。

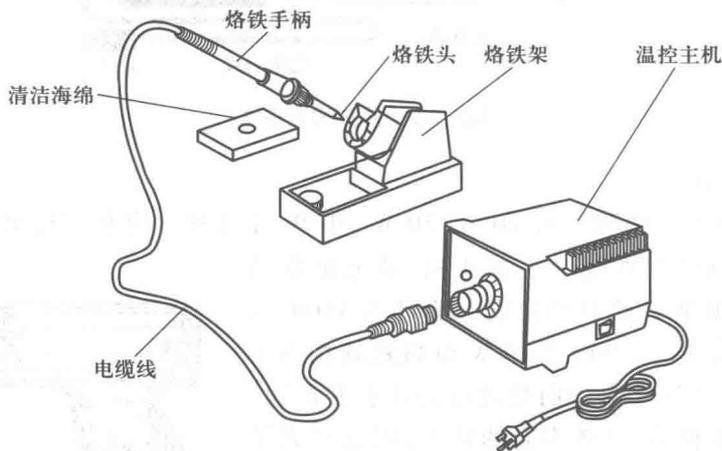


图 6.2.4 温控电烙铁

三、电烙铁的选用

电烙铁的种类及规格有很多种,在使用操作维修时,可根据不同的被焊工件合理地选用电烙铁的功率、种类和烙铁头的形状。一般的焊接应首选内热式电烙铁。对于焊接大型元器件或直

径较粗的导线应选择功率较大的外热式电烙铁。如果被焊件较大,使用的电烙铁功率较小,则焊接温度过低,焊料熔化较慢,焊剂不能挥发,焊点不光滑、不牢固,这样势必造成焊接强度以及质量的不合格,甚至焊料不能熔化,使焊接无法进行。如果电烙铁的功率太大,则使过多的热量传递到被焊工件上面,使元器件的焊点过热,造成元器件的损坏,致使印刷电路板的铜箔脱落,焊料在焊接面上流动过快,并无法控制。

当焊接集成电路、晶体管、受热易损元器件或小型元器件时,应选用 20 W 内热式电烙铁或恒温电烙铁。

当焊接导线及同轴电缆时,应先用 45~75 W 外热式电烙铁,或 50 W 内热式电烙铁。

对一些较大的元器件时,如变压器的引线脚、大电解电容器的引线脚、金属底盘接地焊片或照明电路的连接时,应选用 100 W 以上的电烙铁。

四、电烙铁的使用

1. 电烙铁的正确握法

电烙铁有三种握法:反握式、正握式和握笔式,如图 6.2.5 所示。焊锡丝在连续焊接和断续焊接时拿法也有区别,在拿焊锡丝时,应戴手套或操作后洗手,避免重金属中毒。烙铁架一般放置在工作台右前方,注意导线等物不要碰烙铁头。

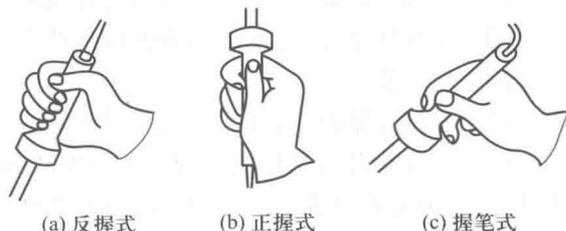


图 6.2.5 电烙铁的握法

2. 烙铁头及修整镀锡

烙铁头的好坏是决定焊接质量和工作效率的重要因素。一般的烙铁头是用纯铜制作的,它的作用是储存和传导热量,它的温度必须比被焊接的材料熔点高。纯铜的润湿性和导热性非常好,但它的一个最大的弱点是容易被焊锡腐蚀和氧化,需要修整和镀锡。方法是将烙铁头拿下夹到台钳上粗锉,再用细锉修平,最后用细砂纸打磨光。修整后将烙铁头装好通电,在木板上放些松香和焊锡,烙铁沾锡后在松香中反复摩擦至均匀镀锡为止,注意,烙铁通电后要立刻蘸上松香,避免氧化。

3. 烙铁头温度的判别和调整

通常情况下,可根据助焊剂的发烟状态直观目测判断烙铁头的温度。如图 6.2.6 所示,在烙铁头上熔化一点松香焊剂,根据助焊剂的发烟量判断其温度是否合适。温度低时,发烟量小,持续时间长;温度高时,发烟量大,消散快;在中等发烟状态,约 6~8 s 消散时,温度约为 300℃ 左右,这时是焊接的合适温度。

烙铁头温度的调整:经过选择电烙铁功率大小后,已基本满足焊接温度的需要,但是仍不能完全适应印刷电路板中所装元器件的需求,比如焊接集成电路和晶体管时烙铁头的温度就不能太高,且时间不能过长,此时便可对烙铁头插在导热管上的长度进

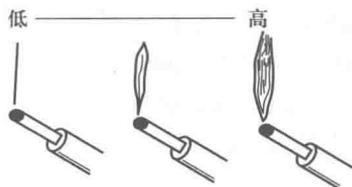


图 6.2.6 烙铁头温度判别

行适当调整,进而控制烙铁头的温度。

4. 电烙铁的使用注意事项

(1) 在使用前或更换烙铁芯后,必须检查电源线与地线的接头是否正确。注意接地线时要正确地接在烙铁的壳体上,如果接错就会造成烙铁外壳带电,人体触及烙铁外壳就会触电,用于焊接则会损坏电路。

(2) 在使用电烙铁的过程中,烙铁电源线不要被烫破,否则可能会使人体触电。应随时检查电烙铁的插头、电源线,发现破损或老化时应及时更换。

(3) 在使用电烙铁的过程中,一定要轻拿轻放,应拿烙铁的手柄部位并且要拿稳。不焊接时,要将烙铁放到烙铁架上,以免灼热的烙铁烫伤自己或他人;长时间不使用时应切断电源,防止烙铁头氧化;不能用电烙铁敲击被焊工件;烙铁头上多余的焊锡,不要随便抛甩,以防落下的焊锡溅到人身上造成烫伤;若溅到正在维修或调试的设备内,焊锡会使设备内部造成短路,造成不应有的损失,要用潮湿的抹布或其他工具将其去除。

(4) 电烙铁在焊接时,最好选用松香或弱酸性助焊剂,以保护烙铁头不被腐蚀。

(5) 经常用湿布、浸水的海绵擦拭烙铁头,以保持烙铁头良好地挂锡,并可防止残留助焊剂对烙铁头的腐蚀。

(6) 焊接完毕时,烙铁头上的残留焊锡应该继续保留,以防止再次加热时出现氧化层。

(7) 人体头部与烙铁头之间一般要保持 30 cm 以上的距离,以避免过多的有害气体吸入体内,因为焊剂加热时挥发出来的化学物质对人体是有害的。

五、焊接材料

1. 焊料

焊料是易熔金属,熔点低于被焊金属,在熔化时在被焊金属表面形成合金从而将被焊金属连接在一起。按焊料成分,有锡铅焊料、银焊料、铜焊料等,在一般电子产品装配中主要使用锡铅焊料。

2. 焊剂

焊剂是一种用于清除氧化膜的专用材料,又称助焊剂。使用助焊剂可以有效去除氧化物,防止氧化,增加焊锡流动性,减小表面张力,使焊点更光亮、美观等。

对助焊剂的要求主要有以下几个方面:① 熔点低于焊料;② 表面张力、黏度、比重小于焊料;③ 残渣易清除;④ 不能腐蚀母材;⑤ 不能产生有害气体和刺激性气体。

助焊剂大致可分为有机焊剂、无机焊剂和树脂焊剂三大类。三者活性从左向右依次减弱,毒性或有害性也依次减弱。其中以松香为主要成分的树脂焊剂在电子产品生产中占有重要地位,成为专用型的助焊剂。

助焊剂加热挥发出来的化学物质对人体有害,所以烙铁离开鼻子的距离至少不少于 30 cm,通常以 40 cm 为宜。

6.3 手工焊接的基本技能

手工焊接是焊接技术的基础,是电子产品装配中的一项基本操作技能。手工焊接适用于小批量电子产品的生产、具有特殊要求的高可靠产品的焊接、某些不便于机器焊接的场所以及调试和维修中的修复焊点和更换元器件等。

一、焊接前的准备

1. 元器件引线弯曲成形

为使元器件在印制电路板上的装配排列整齐并便于焊接,在安装前通常采用手工或专用机械把元器件引脚弯曲成一定的形状。元器件在印制板上的安装方式有三种:立式安装、卧式安装和表面安装。

立式安装和卧式安装无论采用哪种方法,都应该按照元器件在印制电路板上孔位的尺寸要求,使其弯曲成形的引脚能够方便地插入孔内。引脚弯曲处距离元器件实体至少在 2 mm 以上,绝对不能从引线的根部开始弯折,如图 6.3.1 所示。

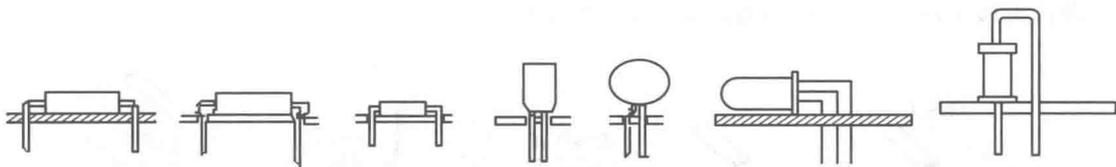


图 6.3.1 元器件引线成形图例

2. 镀锡

为了提高焊接的质量和速度,避免虚焊等缺陷,应该在装配以前对焊接表面进行可焊性处理——镀锡,即在电子元器件的待焊面(引线或其他需要焊接的地方)镀上焊锡。

镀锡时,待镀面应该保持清洁,烙铁头的温度要适合,同时要使用有效的焊剂,比如松香等助焊剂。

二、焊接操作

1. 焊锡丝的拿法

焊锡丝一般有两种拿法,如图 6.3.2 所示。由于在焊锡丝中含有一定比例的铅,而铅又是对人体有害的一种重金属。因此,焊接时应戴上手套或操作后洗手,避免食入铅粉。



(a) 连续送锡

(b) 断续送锡

图 6.3.2 焊锡丝握法

2. 焊接五步法

焊接五步法是常用的基本焊接方法,包括准备施

焊、加热焊件、熔化焊料、移开焊锡、移开烙铁 5 个步骤,适合于焊接热容量大的工件,如图 6.3.3 所示。

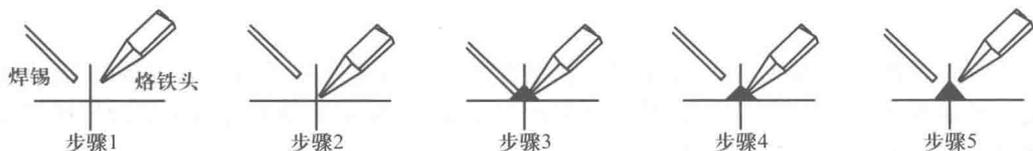


图 6.3.3 焊接五步法

上述过程对一般焊点大约需要二、三秒钟。对于热容量较小的焊点用三步法即可,将 2、3 合为一步,4、5 合为一步。在五步法焊接中要注意准备施焊过程时要保持烙铁头干净,加热焊件要保持焊件各部分均匀受热,熔化过程时间把握恰当,移开焊锡和烙铁的方向大致为斜上 45° 。

在焊接过程中,焊锡用量要适中,如图 6.3.4 所示。



图 6.3.4 焊锡用量

烙铁头撤离也有讲究,撤离方法如图 6.3.5 所示。



图 6.3.5 烙铁头撤离方法

3. 特殊元器件的焊接

(1) 焊接晶体管时,注意每个管子的焊接时间不要超过 10 s,并使用尖嘴钳或镊子夹持管脚散热,以免烫坏晶体管。

(2) 焊接 CMOS 电路时,如果事先已将各引线短路,焊接前不要拿掉短路线,对使用高电压的烙铁,最好在焊接时拔下插头,利用余热焊接。

(3) 焊接集成电路时,在保证浸润的前提下,尽可能缩短焊接时间,一般每个引脚不要超过 2 s。

(4) 焊接集成电路时,电烙铁最好选用 20 W 内热式的,并注意保证良好接地。必要时,还要采取人体接地的措施。

(5) 集成电路若不使用插座直接焊到印制电路板上,安全焊接的顺序是:地端→输出端→电源端→输入端。

4. 导线焊接

导线同接线端子、导线与导线之间的连接有三种基本形式:绕焊、钩焊和搭焊。其中绕焊可靠性最好,常用于要求可靠性高的地方;钩焊的强度低于绕焊,但操作简单;搭焊的连接最方便,但强度及可靠性最差,仅用于临时连接或不便于缠、钩的地方以及某些接插件上。

5. 焊点质量检查

为了保证焊接质量,一般在焊接后都要进行焊点质量检查,主要有以下几种方法。

(1) 外观检查:就是通过肉眼从焊点的外观上检查焊接质量,可以借助3~10倍放大镜进行目检。目检的主要内容有:焊点是否有错焊、漏焊、虚焊和连焊;焊点周围是否有焊剂残留物;焊接部位有无热损伤和机械损伤现象。

(2) 拨动检查:在外观检查中发现有可疑现象时,可用镊子轻轻拨动焊接部位进行检查,并确认其质量。主要包括导线、元器件引线和焊盘与焊锡是否结合良好,有无虚焊现象;元器件引线和导线根部是否有机械损伤。

(3) 通电检查:通电检查必须是在外观检查及连接检查无误后才可进行的工作,也是检验电路性能的关键步骤。如果不经过严格的外观检查,通电检查不仅困难较多,而且容易损坏设备仪器,造成安全事故。通电检查可以发现许多微小的缺陷,例如,用目测观察不到的电路桥接、内部虚焊等。

造成焊接缺陷的原因很多,图6.3.6是接线端子焊接缺陷示例。

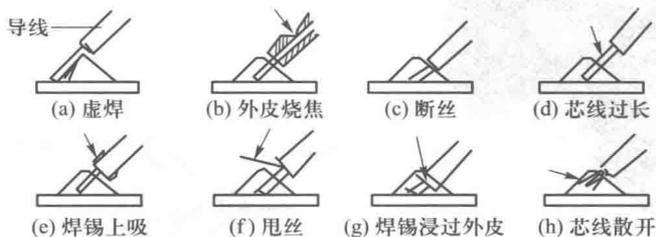


图 6.3.6 导线端子焊接缺陷示例

6.4 拆焊

在调试、维修电子设备的工作中,经常需要更换一些元器件。更换元器件的前提当然是要把原先的元器件拆焊下来。如果拆焊的方法不当,就会破坏印制电路板,也会使换下来但并没失效的元器件无法重新使用。

一、拆焊要点

1. 烙铁头加热被拆焊点,焊料熔化后,要及时按垂直印制电路板的方向拔出元器件的引线。不管元器件的安装位置如何,是否容易取出,都不能强拉或扭转元器件。

2. 在插装新的元器件之前,必须把焊盘插线孔内的焊料清除干净。否则,在插装新元器件引线时,将造成印制电路板的焊盘翘起。

二、拆焊常用方法

1. 采用拆焊专用工具如镊子形烙铁等。如图 6.4.1(a)所示为镊子形烙铁,图 6.4.1(b)所示为镊子形烙铁拆焊方法。

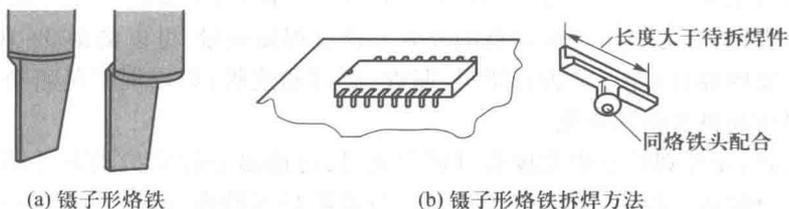


图 6.4.1 采用镊子形烙铁拆焊

2. 采用吸锡泵、吸锡烙铁或吸锡器,如图 6.4.2 所示。

3. 用吸锡材料拆焊,如图 6.4.3 所示。



图 6.4.2 吸锡烙铁

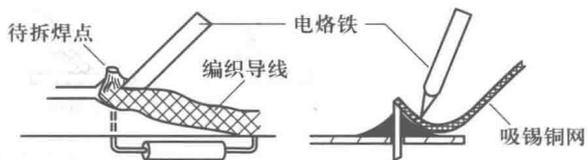


图 6.4.3 用吸锡材料拆焊

4. 清除焊盘插孔内的焊料。

用合适的缝衣针或钢丝,从印制电路板的非焊盘面插入孔内,然后用电烙铁对准焊盘孔加热,待焊料熔化时,缝衣针便从孔中穿出,从而清除了焊孔内的焊料。

第七章

电子技术综合实验硬件电路设计部分

7.1 面包板及其使用

面包板是专为电子电路的无焊接实验设计制造的重要工具。由于各种电子元器件可根据需要随意插入或拔出,免去了焊接,节省了电路的组装时间,而且元件可以重复使用,所以非常适合电子电路的组装、调试和训练。熟练掌握面包板的使用方法是提高实验效率、减少实验故障出现几率的重要基础之一。下面就面包板的结构和使用方法做简单介绍。

一、常用面包板的结构

面包板的外观如图 7.1.1 所示,常见的最小单元面包板分上、中、下三部分,上面和下面部分一般是由一行或两行的插孔构成的窄条,中间部分是由中间一条隔离凹槽和上下各 5 行的插孔构成的宽条。

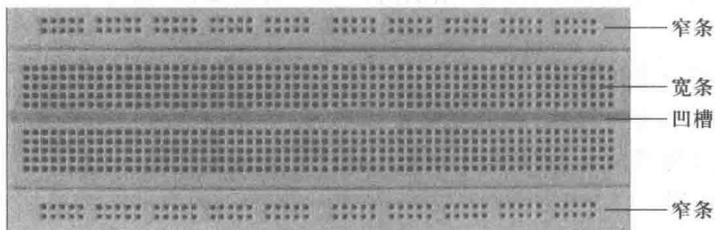


图 7.1.1 面包板的外观

图 7.1.2 所示为窄条部分的外观和结构。

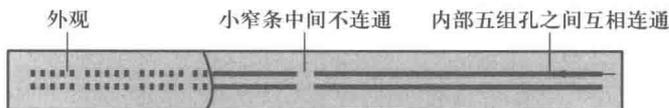


图 7.1.2 窄条部分的外观和结构

窄条上下两行之间电气不连通。每 5 个插孔为一组(通常称为“孤岛”),通常的面包板上有 10 组。这 10 组“孤岛”一般有 3 种内部连通结构:① 左边 5 组内部电气连通,右边 5 组内部电气连通,但左右两边之间不连通,这种结构通常称为 5-5 结构;② 左边 3 组内部电气连通,中间 4 组内部电气连通,右边 3 组内部电气连通,但左边 3 组、中间 4 组以及右边 3 组之间是不连通的,这种结构通常称为 3-4-3 结构;③ 还有一种结构是 10 组“孤岛”都连通,这种结构最简单。

宽条部分是由中间一条隔离凹槽和上下各 5 行的插孔构成。在同一列中的 5 个插孔是互相连通的,列和列之间以及凹槽上下部分则是不连通的。外观及结构如图 7.1.3 所示。

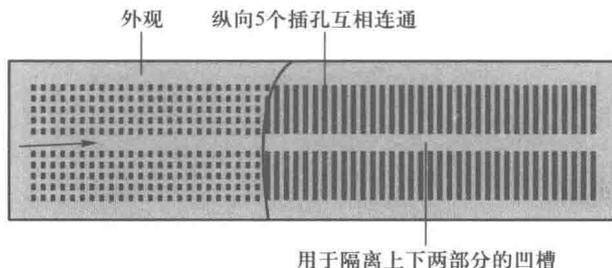


图 7.1.3 宽条部分的外观和结构

在做实验的时候,通常是使用两窄一宽组成的小单元,同学们应按照实验指导教师的示范和要求,在宽条部分搭接电路的主体部分,上面的窄条取一行做电源,下面的窄条取一行做接地。使用时注意窄条的中间部分不通。

二、布线用的工具

布线用的工具主要有剥线钳、偏口钳、扁嘴钳和镊子,参阅本教材第六章第 6.2 节,偏口钳与扁嘴钳配合用来剪断导线和元器件的多余引脚。钳子刃面要锋利,将钳口合上,对着光检查时应合缝不漏光。

剥线钳用来剥离导线绝缘皮。

扁嘴钳用来弯直和理直导线,钳口要略带弧形,以免在勾绕时划伤导线。

镊子是用来夹住导线或元器件的引脚送入面包板指定位置的。

三、面包板的布线

在电路的设计完成之后,就要在面包板上实现设计的电路。在电路设计正确的前提下,布线

对于能否正确实现设计功能是至关重要的。大多数故障是由于布线不正确而引起的。下面简单介绍一些重要的布线原则。

1. 通过外观和通电检查各个芯片及元器件是否完好

2. 芯片(双列直插 DIP 封装的芯片)的放置

(1) 在同一块面包板上,所有芯片的朝向应一致,禁止出现反插。

(2) 对于首次使用的芯片,引脚稍向外偏,使用前用合适的工具齐根轻轻地整成平行,以便于插入面包板的插孔中。

(3) 插芯片时,应放平并均匀用力插入孔中,防止个别引脚未插入而弯曲。

(4) 拔出芯片时,应使用起拔器或小改锥在左右两侧轮流轻撬,平行拔出,避免个别引脚弯曲和断裂。

(5) 对于集成电路中不用的输入端,必须接到一个无效的电平上。

(6) 不用的输出端悬空即可。

3. 布线规则

(1) 一般采用直径为 0.5 mm 的单股圆导线。

(2) 导线的颜色可按用途或功能分类,比如红线接电源,黑线接地等。

(3) 一般情况下,面包板的上边孔用作+5 V,下边孔用作地端。

(4) 导线的预处理:用剥线钳将导线剪成斜面,既便于导线插入面包板,又可以防止划伤面包板内部的弹簧片,然后用剥线钳剥线。

(5) 布线时,导线应紧贴板面,禁止飞线和斜线;导线转角应成 90° ,导线应在芯片周围走线,不得从芯片上跨越;电源线应分在上下两侧,必要时加电源去耦电路,以防止干扰;芯片的电源端就近连接;插线时应尽量不要遮盖插孔。

(6) 掌握连线顺序。先连短线,后连长线,或连接不易连接的导线。特别要注意各集成器件的地线和电源线不要接反,最好先用短线将所有集成电路的地线和电源线接好。

4. 整体布局

(1) 芯片及大型元件布置应疏密得当,使用方便。

(2) 考虑布线的密度,预留空位。

(3) 适当调整芯片的相对位置,使布线合理,尽量减少导线的条数。

(4) 走线要清楚,信号流向清晰,易于检查。

5. 布线检查

当一个单元电路布线完成后,就进行局部检查,以便及时发现故障并排除。检查时,可用万用表的欧姆挡直接测量芯片引脚的通与不通。

四、电子电路的调试

电子电路的调试是整个设计过程的最后一个环节,是检验电路功能尤为重要的一环,在实现的电路设计中,电路的调试是最难最重要的一环,因为如果设计电路不能正常工作,某故障原因很复杂,既可能是布线时的接线错误造成的,也可能是电路设计时考虑问题不全面造成的。因此

为了排除故障,既要熟悉所设计电路的工作原理,又要对电路设计器件的使用有着丰富的知识和调试经验。在此我们简单介绍一下调试的基本原则,调试中的一些经验由同学们在自行调试过程中去总结、摸索和积累。

1. 单元电路的调试

在电路设计时,我们曾把一个整体电路划分为几个单元电路:时钟产生电路、分频电路、计数电路、译码显示电路等,在调试电路时,可以按照信号的流向逐级调试。这样做的好处是比较容易发现故障。假设通电后发现电路不能正常工作,就从时钟产生电路开始,用万用表、示波器逐级测试。先进行外观检查,用万用表检查通电情况,然后再用示波器观察信号波形看哪一级不能正常工作,找出故障的大致范围,然后再仔细分析该局部电路,发现并排除故障。对于数字电路,一般测试顺序如下。

- (1) 定时器。定时器为整个系统提供时钟脉冲和各种定时信号。
- (2) 控制电路。控制电路产生整个系统所需的各种控制信号,包括脉冲分配器、分频器等。
- (3) 信号处理电路。如寄存器、计数器、存储器、编码器、译码器等。
- (4) 模拟电路。如运放、比较器、A/D 转换、D/A 转换等。
- (5) 各种输出执行部件。如发光二极管、灯泡、LED 显示器、蜂鸣器等。

2. 调试内容

(1) 时钟发生电路的调试。用示波器观察此电路的输出波形,根据示波器显示的波形,计算出该时钟信号的频率和周期。

(2) 分频器的调试。用示波器观察分频器的时钟信号和输出波形,观察两级的输出波形是否是十分频的关系。

(3) 计数器电路的调试。将计数器的时钟输入端与电路断开,人为手动加入一个一个脉冲,每次脉冲后,用万用表直流电压挡测量计数器各输出端的电平值,记录下每次脉冲到来前后的输出端的电平高低的变化,也就得到了计数器的计数顺序。判断该计数顺序是否正常。如果正常,再观察本级所产生的进位信号或清零信号是否正常。依次检查其他计数器的工作。

(4) 译码电路的调试。在计数器工作正常的情况下,用万用表蜂鸣挡仔细检查计数器的输出端到译码器的连线,以及译码器的输出端到 LED 数码管之间的连线是否正确。

7.2 数字电子钟的设计、安装与调试

电子钟是一种高精度的计时工具,它采用了集成电路和石英技术,因此走时精度高,稳定性好,使用方便,且不需要经常调校。电子钟根据显示方式不同,分为指针式电子钟和数字式电子钟。指针式电子钟采用机械传动带动指针显示;而数字式电子钟则是采用译码电路驱动数码显示器件,以数字形式显示。这些译码显示器件,利用集成技术可以做得非常小巧,也可以另加一定的驱动电路,推动霓虹灯或白炽灯显示系统,制作成大型电子钟表。因此,数字式电子钟用

途非常广泛。

一、设计任务与要求

(1) 设计一台能直接显示时、分、秒的数字电子钟。小时可采用十二进制也可采用二十四进制。

(2) 设计校“时”、校“分”的控制电路。

(3) 设计 24 小时整点报时控制电路,报时时,声响共五响,最后一响为整点。

(4) 根据规定的作息时间表,如表 7.2.1 所示,设计自动响铃控制电路。

表 7.2.1 作息时间表

作息时间表	
起床	6 : 50
上午上班	8 : 00
午饭	11 : 45
下午上班	13 : 30
下班	17 : 30

二、总体方案设计

数字式电子钟的基本功能是能够实现时、分、秒的正确计时,计时单位为 1 秒。因此,一个简单的数字式电子钟,首先必须有计时显示电路和秒脉冲产生电路。其次,当刚接通电源或时钟走时出现误差时,需要进行时间校准,否则就不能正确表示当前时间。因此,数字式电子钟应有校时控制电路。另外,若要求数字钟能够自动整点报时或按要求时间闹铃,还应有整点报时和闹铃控制电路。若还需要其他功能,相应的还要有一些控制电路。综上所述,数字式电子钟应由三大部分组成:即计时显示电路、秒脉冲产生电路和控制电路。其结构框图如图 7.2.1 所示。

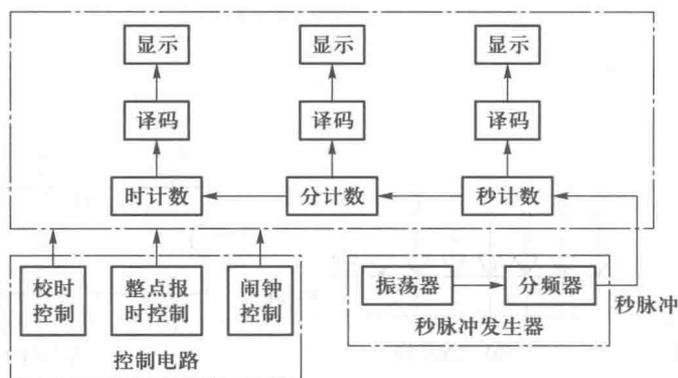


图 7.2.1 数字式电子钟的结构框图

部分电路如下。

(1) 七段显示数码管和七段显示译码器,如图 7.2.2 所示。

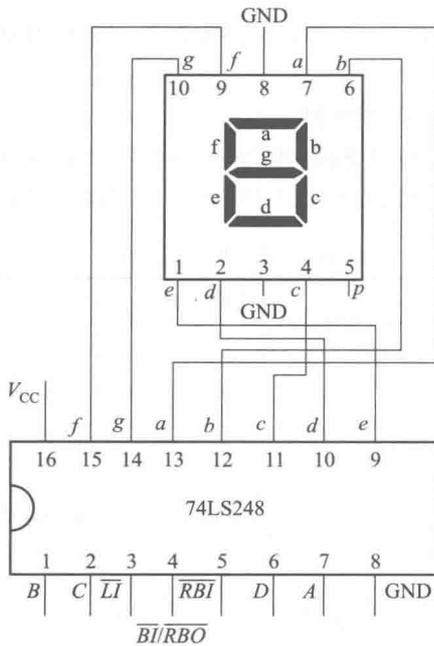


图 7.2.2 七段显示数码管和七段显示译码器

共阴极七段显示数码管的符号如图 7.2.3(a) 所示,它的内部结构如图 7.2.3(b) 所示。公共端接低电平,当 $a \sim h$ 某一端为高电平时,相应的发光二极管发光。

74LS248 是共阴极数码管 BCD-七段译码器,引脚分布图如图 7.2.4 所示。表 7.2.2 是 74LS248 的功能表。

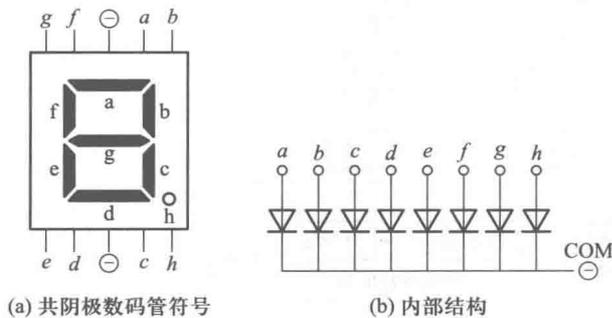


图 7.2.3 七段显示译码器

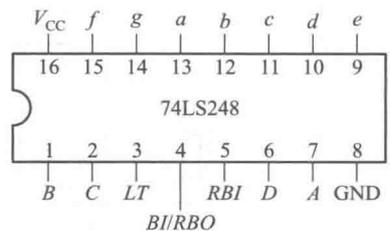


图 7.2.4 七段显示译码器

表 7.2.2 74LS248 的功能表

功能	输入						输入/输出	输出							显示字形
	\overline{LT}	\overline{RBI}	D	C	B	A		$\overline{BI}/\overline{RBO}$	a	b	c	d	e	f	
试灯	0	×	×	×	×	×	1(出)	1	1	1	1	1	1	1	消隐
灭零	1	0	0	0	0	0	0(出)	0	0	0	0	0	0	0	消隐
灭灯	×	×	×	×	×	×	0(入)	0	0	0	0	0	0	0	消隐
0	1	1	0	0	0	0	1(出)	1	1	1	1	1	1	0	0
1	1	×	0	0	0	1	1(出)	0	1	1	0	0	0	0	1
2			0	0	1	0	1(出)	1	1	0	1	1	0	1	2
3			0	0	1	1	1(出)	1	1	1	1	0	0	1	3
4			0	1	0	0	1(出)	0	1	1	0	0	1	1	4
5			0	1	0	1	1(出)	1	0	1	1	0	1	1	5
6			0	1	1	0	1(出)	0	0	1	1	1	1	1	6
7			0	1	1	1	1(出)	1	1	1	0	0	0	0	7
8			1	0	0	0	1(出)	1	1	1	1	1	1	1	8
9			1	0	0	1	1(出)	1	1	1	0	0	1	1	9

(2) 10 MHz 振荡器和十分频电路,如图 7.2.5 所示。

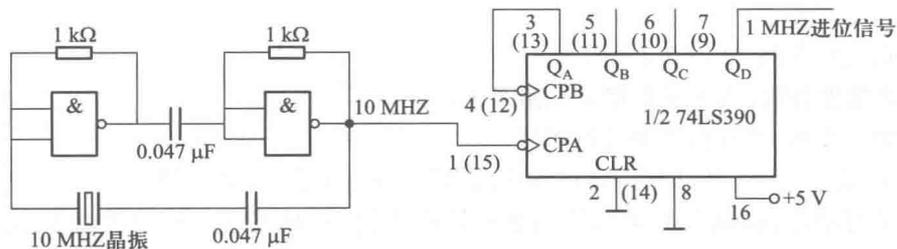


图 7.2.5 十分频电路

74LS390 是二-五-十进制计数器,功能表如表 7.2.3 所示。

表 7.2.3 74LS390 的功能表

CLR	CP	Q_D	Q_C	Q_B	Q_A
1	×	0	0	0	0
0	1	0	0	0	1
0	2	0	0	1	0

续表

CLR	CP	Q_D	Q_C	Q_B	Q_A
0	3	0	0	1	1
0	4	0	1	0	0
0	5	0	1	0	1
0	6	0	1	1	0
0	7	0	1	1	1
0	8	1	0	0	0
0	9	1	0	0	1

(3) 校时电路,如图 7.2.6 所示。



图 7.2.6 校时电路

(4) 整点报时电路,如图 7.2.7 所示。

三、设计报告要求(设计报告格式见附录 F)

(1) 目的与要求:根据任务书进行填写。

(2) 正文:总体设计、各部分设计及器件的选用;电路框图;调试步骤。

(3) 总结与结论:调试中所遇到的问题及解决方案;对最终成品的自我评价;心得体会和设想。

四、实验仪器及主要器件

(1) 仪器仪表

SDS1072CML 型数字存储示波器 1 台

直流稳压电源 1 台

数字万用表 1 只

(2) 元器件清单

74LS390×7, 74LS248×6, 555×1, 74LS74×1, 74LS30×1, 74LS00(若干), 10 MHz 晶振×1, 共阴极七段数码管×6, 蜂鸣器×1。

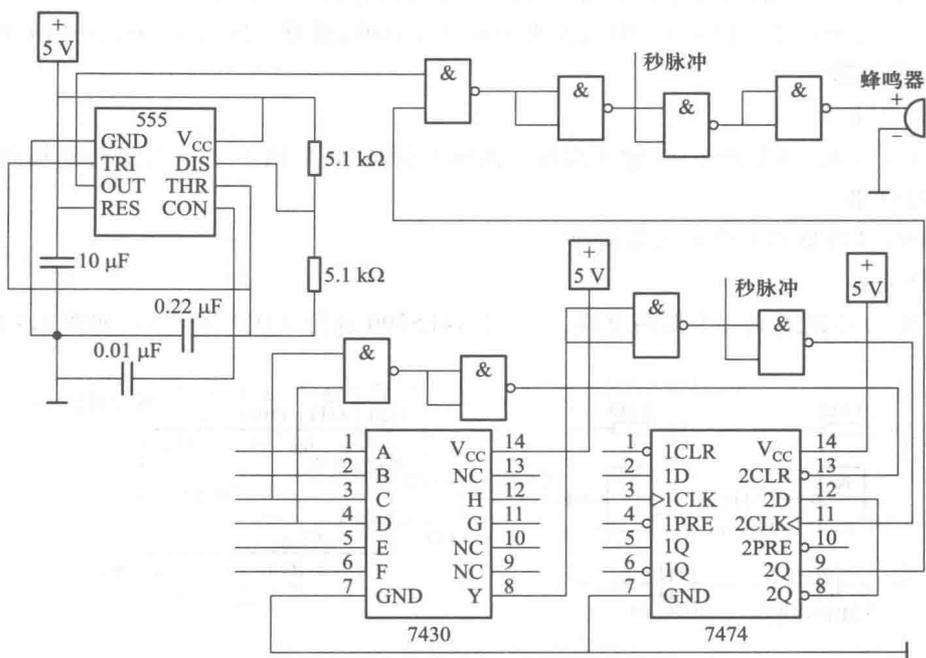


图 7.2.7 整点报时电路

7.3 智力竞赛抢答器设计、安装与调试

在进行智力竞赛抢答题比赛时,各参赛者考虑好后都想抢先回答,如果没有合适的设备,主持人难以分清抢答者的先后。为了使比赛能顺利进行,需要有一个能判断抢答者先后的设备,称其为智力竞赛抢答器。

一、设计任务与要求

(1) 此抢答器可以容纳六组参赛队,每组设置一个抢答按钮供抢答者使用。设置一个“系统复位”或“抢答准备命令”按钮和一个“抢答开始命令”按钮供主持人使用。

(2) 电路具有第一抢答信号的鉴别和锁存功能。在主持人将系统复位并发出“抢答开始命令”后,若参赛者按下抢答按钮,就显示最先抢答者的组号,指示抢答有效,并以声音警示。若系统复位但未发“抢答开始命令”,参赛者就按下抢答按钮,也显示抢答者的组号,但是抢答无效,并以声音警示。要求确定第一个输入的抢答信号,并保持该信号不变,同时使后输入的信号无效。

(3) 在发出“抢答开始命令”后开始计时,经过规定的抢答时间后若没有人抢答,就发出“抢

答时间到”信号,以声光警示,并锁定输入电路使各路抢答信号无法再输入。

(4) 设置计分电路,开始时每组预置为 100 分或其他,答对一次加 10 分,答错减 10 分。

二、实验原理

1. 抢答电路

由六组同步 RS 触发器构成,输入端接一两输入的与非门,用 1~6 个开关分别控制。

2. 计时电路

由 74160 计数器和七段显示器构成。

3. 秒脉冲

由 10 MHz 晶振组成一个振荡电路,由 4 个 74LS390 进行 4 次分频得到,如图 7.3.1 所示。

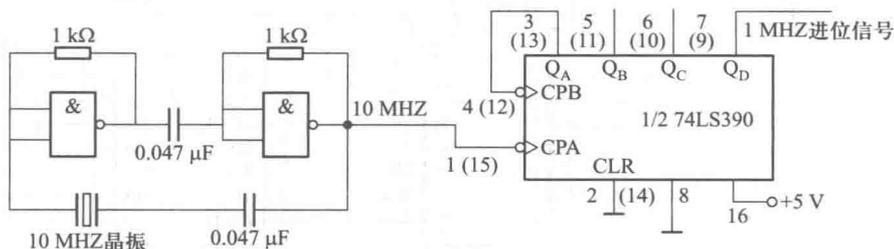


图 7.3.1 秒脉冲产生电路

4. 译码显示电路

七段译码显示电路如图 7.3.2 所示。

三、设计报告要求(设计报告格式见附录 F)

(1) 目的与要求:根据任务书进行填写。

(2) 正文:总体设计、各部分设计及器件的选用;电路框图;调试步骤。

(3) 总结与结论:调试中所遇到的问题及解决方案;对最终成品的自我评价;心得体会和设想。

四、实验仪器及主要器件

1. 仪器仪表

SDS1072CML 型数字存储示波器 1 台

直流稳压电源 1 台

数字万用表 1 只

2. 元器件清单

10 MHz 晶振×1, 0.047 μF 电容×2, 1 kΩ 电阻×2, 74LS00×4, 74LS390×4, 74LS27×1, 74LS08×2, 74LS02×1, 74LS04×2, 74LS10×1, 74LS160×1, 74LS74×1, 74LS248×2,

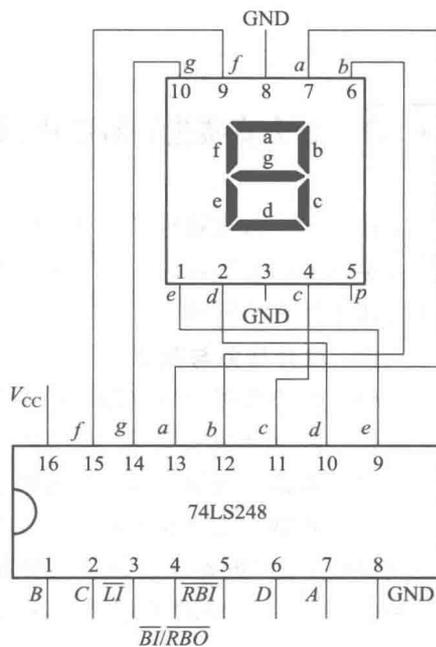


图 7.3.2 七段译码显示电路

共阴极数码管×2,红、绿、黄发光二极管各×1。

7.4 电子秒表的设计、安装与调试

电子秒表是一种较先进的电子计时器。本章我们的设计是基于数字电路和模拟电路的设计思路及实现方法。主要是以 555 定时器作为发生器,以分频、计数与译码显示模块为主要构成部分;通过电源清零电路和反馈清零电路实现清零功能,构成复位电路;利用启动开关和停止开关控制触发器产生启动/停止信号,实现秒表的启动和停止功能,构成控制电路;再经过面包板布线、安装与调试等工作构成一个简易的电子秒表。

一、设计任务与要求

(1) 利用 555 定时器制作一个时钟发生装置,电路如图 7.4.1 所示。试分析该电路的工作原理,计算时钟脉冲的周期和频率,并画出该电路的工作波形。

(分析)

(2) 通过分频电路,输出周期为 0.01 s 的计数脉冲。(设计)

(3) 利用计数器接收分频电路输出的计数脉冲,并通过数码显示器显示出来,完成 0.01~9.99 s 的秒表计时。(设计)

(4) 该系统具有清零、计时和停止三种功能。(设计)

(5) 采用 Multisim 画出整体电路图,并在面包板上安装和调试。(动手实践)

二、总体方案和单元电路的设计

1. 总体方案设计框图如图 7.4.2 所示:



图 7.4.2 总体方案设计框图

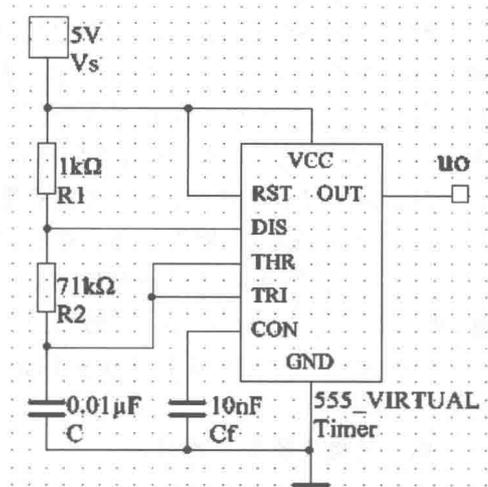


图 7.4.1 555 定时器构成的时钟发生电路

2. 单元电路方案设计

(1) 555 时钟发生模块

利用 555 定时器实现的多谐振荡电路能够完成时钟信号发生器的功能,通过电路中元器件的参数以及电路的工作原理,可以计算出该电路的振荡周期和频率。

(2) 分频电路模块

利用 74LS390(二-五-十进制计数器)将 555 定时器构成的时钟发生器的输出进行分频变为 100 Hz 的信号输出,即周期为 0.01 s。

(3) 输出及显示模块

三个 74LS290 分别连接成十进制计数器,一个输出十分之一秒,一个输出整秒个位,一个输出整秒十位,并通过三个数码显示器显示 0.01~9.99 s 的秒表数值。

(4) 控制电路

利用基本 RS 触发器生成控制电路: S 有效,则 Q 端输出高电平,控制时钟信号输出到分频电路,实现“开始计数”;同时 \bar{Q} 端输出的低电平使“清零并准备重新开始计数”无效; R 有效,则时钟信号被屏蔽,电子秒表保持当前数值不变,同时允许“清零并准备重新开始计数”信号输入。注意,设计时要避免基本 RS 触发器出现不定态。

三、设计报告要求(设计报告格式见附录 F)

(1) 目的与要求:根据任务书进行填写。

(2) 正文:简单叙述设计过程(包括原理、方案);画出完整的电路原理图,简述各部分的功能;画出时钟电路、分频电路、计数电路的时序图等。

(3) 总结与结论:写出设计和调试过程中出现的问题及解决方法;简述心得体会及改进建议。

四、实验仪器及主要器件

(1) 仪器仪表

SDS1072CML 型数字存储示波器 1 台

直流稳压电源 1 台

数字万用表 1 只

(2) 面包板一块

(3) 工具:镊子、剥线钳、一字改锥 各 1 把

(4) 导线:红、黑、白、蓝等若干

(5) 集成电路

555 定时器×1, 74LS00(四-二输入与非门)×1, 74LS04(六反相器)×1, 74LS390(双 4 位异步十进制计数器)×3, 74LS248(BCD 七段译码器)×3, 共阴极数码管×3, 1 kΩ 电阻×3, 51 kΩ 电阻×1, 20 kΩ 电阻×1, 0.01 μF 电容×2, 按键开关×3。

附录 A

集成电路简介

一、集成电路介绍

集成电路(integrated circuit,简称 IC)是一种微型电子器件或部件。采用一定的工艺,把一个电路中所需的晶体管、二极管、电阻、电容和电感等元件及布线互连一起,制作在一小块或几小块半导体晶片或介质基片上,然后封装在一个管壳内,成为具有所需电路功能的微型结构;其中所有元件在结构上已组成一个整体,使电子元件向着微小型化、低功耗和高可靠性方面迈进了一大步。

集成电路按其功能、结构的不同,可以分为模拟集成电路、数字集成电路和数/模混合集成电路三大类。集成电路按集成度可分为小规模、中规模、大规模和超大规模等。小规模集成电路(SSI)是在一块硅片上制成约 1~10 个门,通常为逻辑单元电路,如逻辑门、触发器等。中规模集成电路(MSI)的集成度约为 10~100 门/片,通常是逻辑功能电路,如译码器、数据选择器、计数器、寄存器等。大规模集成电路(LSI)的集成度约为 100 门/片以上,超大规模(VLSI)约为 1 000 门/片以上,通常是一个小的数字逻辑系统。

数字集成电路还可分为双极型电路和单极型电路。双极型电路中有代表性的是 TTL 电路;单极型电路中有代表性的是 CMOS 电路。国产 TTL 集成电路的标准系列为 CT54/74 系列,其功能和外引线排列与国际 54/74 系列相同。国产 CMOS 集成电路主要为 CC(CH)4000 系列,其功能和外引线排列与国际 CD4000 系列相对应。

必须正确了解集成电路参数的意义和数值,并按规定使用。特别是必须严格遵守极限参数的限定,因为即使瞬间超出,也会使器件遭受损坏。

集成电路都有两个或三个电源接线端:用 V_{CC} 、 V_{DD} 、 V_{SS} 、 $+V$ 、 $-V$ 或 GND 来表示。

二、数字集成电路封装

中、小规模数字 IC 中最常用的是 TTL 电路和 CMOS 电路。TTL 器件型号以 74(或 54)作前缀,称为 74/54 系列,如 74LS10、74LS181、54S86 等。中、小规模 CMOS 数字集成电路主要是 4XXX/45XX(X 代表 0~9 的数字)系列,高速 CMOS 电路 HC(74HC 系列),与 TTL 兼容的高速 CMOS 电路 HCT(74HCT 系列)。TTL 电路与 CMOS 电路各有优缺点,TTL 速度快,CMOS 电路功耗小、电源范围大、抗干扰能力强。由于 TTL 在世界范围内应用很广,所以在数字电路教学实验中,我们主要使用 TTL74 系列电路作为实验用器件,采用单一+5 V 作为供电电源。

数字 IC 器件有多种封装形式,如附录图 A.1 所示。最早的集成电路是提供给军方使用的,处于稳定性的考虑都采用陶瓷双列直插封装(CDIP,附录图 A.1(a));当集成电路大规模商业化之后,很快出现了用塑料双列直插封装(PDIP,附录图 A.1(b));到了 20 世纪 80 年代,出现了超大规模集成电路,采用针格阵列封装(PGA,附录图 A.1(c))和无引线芯片载体封装(LCC,附录图 A.1(d)),许多台式机的 CPU 都是 PGA 封装;80 年代初还出现了表面贴元器件(SMD),体积很小,其金属管脚呈海鸥翅膀型或 J 型,且管脚间距只有 DIP 的一半,初期的表面贴元器件采用小轮廓封装(SOIC,附录图 A.1(e))和塑料引线型载体封装(PLCC,附录图 A.1(f)),到了 90 年代末,表面贴元器件又出现了塑料四方扁平封装(PQFP,附录图 A.1(g))和塑型薄片式封装(TSOP,附录图 A.1(h)),这两种封装在多管脚器件如单片机、DSP、存储器中应用非常普遍,但是高端处理器仍然以 PGA 封装为主。此外,还有一种更小的薄小外形封装(TSSOP,附录图 A.1(i)),常常应用在便携、低功耗的产品中。

同一种型号的集成电路可以有一种或多种封装,为了教学实验方便,实验中选用塑料双列直插封装的 74 系列集成电路。双列直插封装有以下特点。

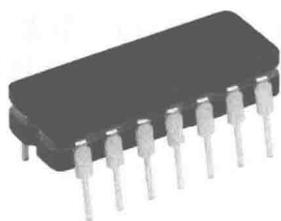
(1) 从正面(上面)看,器件一端有一个半圆的缺口,这是正方向的标志。缺口下边的引脚号为 1,引脚号按逆时针方向增加。附录图 A.2(b)中的数字表示引脚号。双列直插封装 IC 引脚数有 14、16、20、24、28 等若干种。

(2) 双列直插器件有两列引脚。引脚之间的间距是 2.54 mm。两列引脚之间的距离有宽(15.24 mm)、窄(7.62 mm)两种。两列引脚之间的距离能够稍做改变,引脚间距不能改变。将器件插入实验板上的插座中去,从插座中拔出时要小心,不要将器件引脚弯曲或折断。

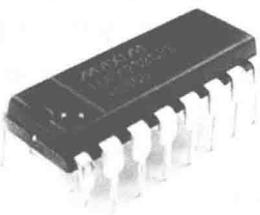
(3) 74 系列器件一般左下角的最后一个引脚是 GND,右上角的引脚是 V_{CC} 。例如,14 引脚器件引脚 7 是 GND,引脚 14 是 V_{CC} ;20 引脚器件引脚 10 是 GND,引脚 20 是 V_{CC} 。但也有一些例外,例如 16 引脚的双 JK 触发器 74LS76,引脚 13(不是引脚 8)是 GND,引脚 5(不是引脚 16)是 V_{CC} 。所以使用集成电路器件时要先看清它的引脚图,找对电源和地,避免因接线错误造成器件损坏。同时还必须注意,不能带电插、拔器件。插、拔器件只能在关断电源的情况下进行。

三、集成电路管脚的判别

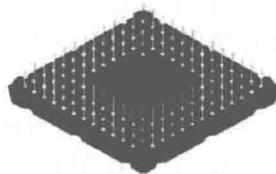
集成电路的管脚都在 3 个以上,在使用时就存在一个管脚判别的问题。通常情况可以参阅集成电路的数据手册,数据手册中会给出管脚排列图,并会给出相应管脚的功能,如附录图 A.2(a)所示。



(a) CDIP封装



(b) PDIP封装



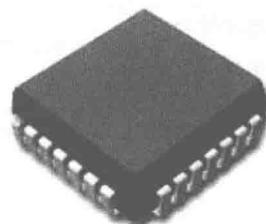
(c) PGA封装



(d) LCC封装



(e) SOIC封装



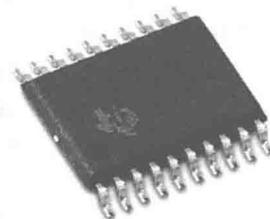
(f) PLCC封装



(g) PQFP封装

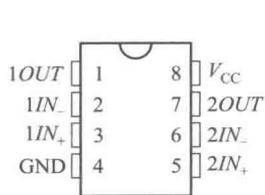


(h) TSOP封装

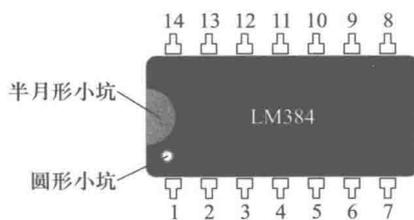


(i) TSSOP封装

附录图 A.1 集成电路的封装形式



(a) 管脚排列图



(b) 集成电路管脚的分布

附录图 A.2 集成电路管脚的判别

对于直插型或表面贴的实际器件来说,外壳上都有一个半月形小坑或圆形小坑,它们是可以用于帮助定位 1 管脚。正对着器件表面的型号,半月形小坑或圆形小坑下方为 1 号管脚,如附图 A.2(b)所示,按逆时针方向,依次为 2 号、3 号、…、 n 号管脚。把这个顺序与技术手册中的管脚排列图对应起来就知道每个管脚的功能了。

四、集成电路的命名方法

集成电路的命名方法见附录表 A.1。

附录表 A.1 国产半导体集成电路型号命名法(GB3430-82)

第零部分		第一部分		第二部分	第三部分		第四部分	
表示器件符合国家标准		表示器件的类型		表示器件的系列和品种代号	表示器件的工作温度范围		表示器件的封装形式	
符号	意义	符号	意义		符号	意义	符号	意义
C	中国制造	T	TTL		C E R M · · ·	0~70℃ -48℃~75℃ -55℃~85℃ -55℃~125℃	W	陶瓷封装
		H	HTL				B	塑料扁平
		E	ECL				F	全密封扁平
		C	CMOS				D	陶瓷直插
		F	线性放大器				P	塑料直插
		D	音响、电视				J	黑陶瓷扁平
		W	电路稳压器				K	金属菱形
		J	接口电路				T	金属圆形
		B	非线性电路					
		M	存储器					
μ	微型电路							

附录 B

集成逻辑电路的连接和驱动

一、TTL 电路输入输出电路性质

当输入端为高电平时,输入电流是反向二极管的漏电流,电流极小。其方向是从外部流入输入端。

当输入端处于低电平时,电流由电源 V_{CC} 经内部电路流出输入端,电流较大,当与上一级电路衔接时,将决定上级电路应具有负载能力。高电平输出电压在负载不大时为 3.5 V 左右。低电平输出时,允许后级电路灌入电流,随着灌入电流的增加,输出低电平将升高,一般 LS 系列 TTL 电路允许灌入 8 mA 电流,即可吸收后级 20 个 LS 系列标准门的灌入电流。最大允许低电平输出电压为 0.4 V。

二、CMOS 电路输入输出电路性质

一般 CC 系列的输入阻抗可高达 $10^{10} \Omega$,输入电容在 5 pF 以下,输入高电平通常要求在 3.5 V 以上,输入低电平通常为 1.5 V 以下。因 CMOS 电路的输出结构具有对称性,故对高低电平具有相同的输出能力,负载能力较小,仅可驱动少量的 CMOS 电路。当输出端负载很轻时,输出高电平将十分接近电源电压;输出低电平时将十分接近地电位。

在高速 CMOS 电路 54/74HC 系列中的一个子系列 54/74HCT,其输入电平与 TTL 电路完全相同,因此在相互取代时,不需考虑电平的匹配问题。

三、集成逻辑电路的连接

在实际的数字电路系统中总是将一定数量的集成逻辑电路按需要前后连接起来。这时,前级电路的输出将与后级电路的输入相连并驱动后级电路工作。这就存在着电平的配合和负载能

力这两个需要妥善解决的问题。

可用下列几个表达式来说明连接时所要满足的条件:

$$V_{OH}(\text{前级}) \geq V_{IH}(\text{后级});$$

$$V_{OL}(\text{前级}) \leq V_{IL}(\text{后级});$$

$$I_{OH}(\text{前级}) \geq n \times I_{IH}(\text{后级});$$

$$I_{OL}(\text{前级}) \geq n \times I_{IL}(\text{后级}); \text{其中 } n \text{ 为后级门的数目。}$$

1. TTL 与 TTL 的连接

TTL 集成逻辑电路的所有系列,由于电路结构形式相同,电平配合比较方便,不需要外接元件可直接连接,不足之处是受低电平时负载能力的限制。附录表 B.1 列出了 74 系列 TTL 电路的扇出系数。

附录表 B.1 74 系列 TTL 电路的扇出系数

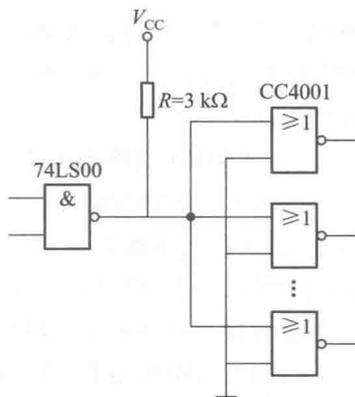
型号	74LS00	74ALS00	7400	74L00	74S00
74LS00	20	40	5	40	5
74ALS00	20	40	5	40	5
7400	40	80	10	40	10
74L00	10	20	2	20	1
74S00	50	100	12	100	12

2. TTL 驱动 CMOS 电路

TTL 电路驱动 CMOS 电路时,由于 CMOS 电路的输入阻抗高,故此驱动电流一般不会受到限制,但在电平配合问题上,低电平是可以的,高电平时有困难,因为 TTL 电路在满载时,输出高电平通常低于 CMOS 电路对输入高电平的要求,因此为保证 TTL 输出高电平时后级的 CMOS 电路能可靠工作,通常要外接一个提拉电阻 R ,如附录图 B.1 所示,使输出高电平达到 3.5V 以上, R 的取值为 2~6.2 k Ω 较合适,这时 TTL 后级的 CMOS 电路的数目实际上是没有什麼限制的。

3. CMOS 驱动 TTL 电路

CMOS 的输出电平能满足 TTL 对输入电平的要求,而驱动电流将受限制,主要是低电平时的负载能力。附录表 B.2 列出了一般 CMOS 电路驱动 TTL 电路时的扇出系数,从表中可见,除了 74HC 系列外的其他 CMOS 电路驱动 TTL 的能力都较低。



附录图 B.1 TTL 电路驱动 CMOS 电路

附录表 B.2 一般 CMOS 电路驱动 TTL 电路时的扇出系数

型号	LS-TTL	L-TTL	TTL	ASL-TTL
CC4001B 系列	1	2	0	2
MC14001B 系列	1	2	0	2
MM74HC 及 74HCT 系列	10	20	2	20

既要使用此系列又要提高其驱动能力时,可采用以下两种方法。

(1) 采用 CMOS 驱动器,如 CC4049、CC4050 是专为给出较大驱动能力而设计的 CMOS 电路。

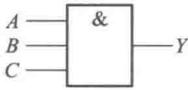
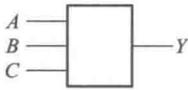
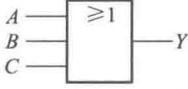
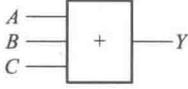
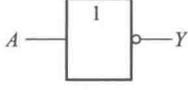
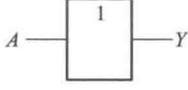
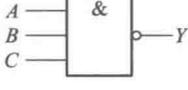
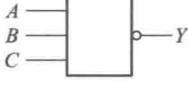
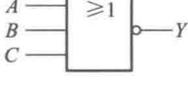
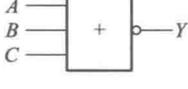
(2) 几个同功能的 CMOS 电路并联使用,即将其输入端并联,输出端并联(TTL 电路是不允许并联的)。

4. CMOS 与 CMOS 的连接

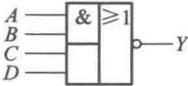
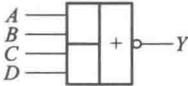
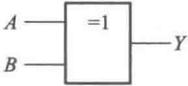
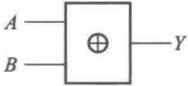
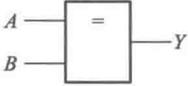
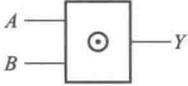
CMOS 电路之间的连接十分方便,不需另加外接元件。对直流参数来讲,一个 CMOS 电路可带动的 CMOS 电路数量是不受限制的,但在实际使用时,应当考虑后级门输入电容对前级门的传输速度的影响,电容太大时,传输速度要下降,因此在高速使用时要从负载电容来考虑,例如 CC4000T 系列。CMOS 电路在 10 MHz 以上速度运用时应限制在 20 个门以下。

附录 C

集成逻辑门电路新、旧图形符号对照

名称	新国标图形符号	旧图形符号	逻辑表达式
与门			$Y = ABC$
或门			$Y = A + B + C$
非门			$Y = \bar{A}$
与非门			$Y = \overline{ABC}$
或非门			$Y = \overline{A + B + C}$

续表

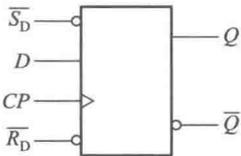
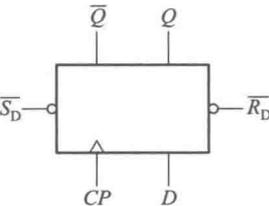
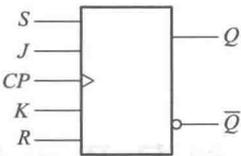
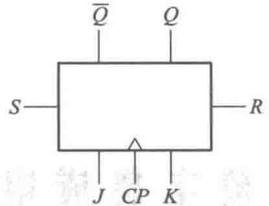
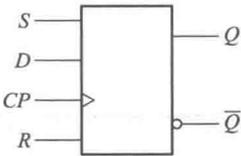
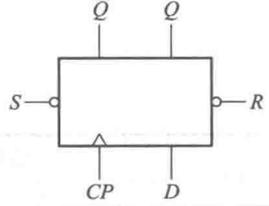
名称	新国标图形符号	旧图形符号	逻辑表达式
与或非门			$Y = \overline{AB + CD}$
异或门			$Y = A\bar{B} + \bar{A}B$
同或门			$Y = AB + \bar{A}\bar{B}$

附录D

集成触发器新、旧图形符号对照

名称	新国标图形符号	旧图形符号	触发方式
由与非门构成的基本RS触发器			无时钟输入, 触发器状态直接由 R 、 S 的电平控制
由或非门构成的基本RS触发器			
TTL 边沿型 JK 触发器			CP 脉冲下降沿

续表

名称	新国标图形符号	旧图形符号	触发方式
TTL 边沿型 D 触发器			CP 脉冲上升沿
CMOS 边沿型 JK 触发器			CP 脉冲上升沿
CMOS 边沿型 D 触发器			CP 脉冲上升沿

附录 E

常用数字集成电路型号及引脚图

电路名称及符号	引脚图	注释
四二输入与非门 74LS00		A、B:输入 Y:输出
四二输入与非门 (OC 门) 74LS01		A、B:输入 Y:输出
四二输入或非门 74LS02		A、B:输入 Y:输出

续表

电路名称及符号	引脚图	注释
六反相器 74LS04		A: 输入 Y: 输出
双四输入与非门 74LS20		NC: 空脚 A、B、C、D: 输入 Y: 输出
八输入与非门 74LS30		NC: 空脚 A~H: 输入 Y: 输出
四线-十线译码器 74LS42		A: 输入 Y: 输出 (低电平有效)
双 JK 触发器 74LS73		下降沿触发
双 D 型触发器 74LS74		上升沿触发

续表

电路名称及符号	引脚图	注释
四二输入异或门 74LS86		A、B:输入 Y:输出
双 JK 触发器 74LS112		下降沿触发
四三态输出门 74LS125		A:输入 Y:输出 \overline{EN} :使能端
3 线-8 线译码器 74LS138		$\overline{G_{2A}}$ 、 $\overline{G_{2B}}$ 、 G_1 :控制端
十线-四线优先编码器 74LS147		\overline{I} :输入 \overline{Y} :输出 (低电平有效) NC:空脚
双四选一数据选择器 74LS153		\overline{S} :控制输入端 D:数据输入端 Y:输出端

续表

电路名称及符号	引脚图	注释
十进制同步加计数器 74LS160		EP 、 ET :计数控制端 \overline{LD} :同步置数端 $\overline{R_D}$:异步清零端 RCO :进位输出端
4位二进制加计数器 74LS161		EP 、 ET :计数控制端 \overline{LD} :同步置数端 $\overline{R_D}$:异步清零端 RCO :进位输出端
四D触发器 74LS175		\overline{MR} :异步清零端 D_{0-3} :数据输入端 Q_{0-3} 、 \overline{Q}_{0-3} :互补输出端
同步可逆十进制计数器 74LS192		$CP_+ = 1$ $CP_- = \uparrow$ 减法 $CP_+ = \downarrow$ $CP_- = 1$ 加法
4位双向移位寄存器 74LS194		\overline{CR} :异步清零端 D_{0-3} :数据输入端 Q_{0-3} :数据输出端 S_1 、 S_0 :方式控制端 D_{SR} :右移数据输入端 D_{SL} :左移数据输入端

续表

电路名称及符号	引脚图	注释
共阴极数码管 BCD-七段译码器 74LS248	<p style="text-align: center;">74LS248</p>	A, B, C, D : 输入 a, b, c, d, e, f, g : 输出, 高电平有效
四位超前进位加法器 74LS283	<p style="text-align: center;">74LS283</p>	A, B : 被加数, 加数 C : 进位端 F : 和
二-五-十进制 异步计数器 74LS290	<p style="text-align: center;">74LS290</p>	$R_{0(1)}, R_{0(2)}$: 清零端 $S_{9(1)}, S_{9(2)}$: 置9端 Q_A, Q_B, Q_C, Q_D : 输出端 NC: 使能端
双二-五-十进制 异步计数器 74LS390	<p style="text-align: center;">74LS390</p>	CLR : 异步清零 A, B : 时钟输入端 Q_A, Q_B, Q_C, Q_D : 输出端
555 定时器	<p style="text-align: center;">555</p>	\overline{TR} : 触发输入端 U_O : 输出端 R_D : 复位段 U_{IC} : 控制电压端 TH : 阈值输入端 U'_O : 放电端
556 定时器	<p style="text-align: center;">556</p>	\overline{TR} : 触发输入端 U_O : 输出端 R : 复位段 U_{IC} : 控制电压端 TH : 阈值输入端 U'_O : 放电端

续表

电路名称及符号	引脚图	注释
ADC 0809		<p>IN_{0-7}: 模拟信号输入端</p> <p>A_{0-2}: 地址输入端</p> <p>ALE: 地址锁存允许输入端</p> <p>$START$: 启动信号输入端</p> <p>EOC: 转换结束输出端</p> <p>OE: 输入允许信号</p>
DAC 0832		<p>\overline{CS}: 片选信号</p> <p>\overline{WR}_{1-2}: 写信号</p> <p>D_{0-7}: 数字信号输入端</p> <p>V_{REF}: 基准电压</p> <p>R_{FB}: 反馈电阻</p> <p>ILE: 输入寄存器允许</p> <p>\overline{XFER}: 传送控制信号</p> <p>I_{out1-2}: 电流输出端</p>

附录F

综合实验的设计报告

设计报告是把设计的目的、方法、过程、结果等记录下来,经过整理写成的书面材料。

设计报告必须在科学实验的基础上进行。它主要的用途在于帮助设计者不断地积累研究资料,总结研究成果。因此,写设计报告是一件非常严肃、认真的工作。不允许草率、马虎,哪怕是一个小数点、一个细微的变化,都不能忽视。

电子技术综合实验的设计报告大体上按照设计步骤和顺序来写,主要包括封面、目录、任务书、正文、附录和参考文献六个部分组成。

1. 封面

封面单独一页,按照设计报告的提示写上课程名称、所选设计题目的名称、所在院系、班级、学号、姓名、指导教师、设计周数以及撰写报告的时间等相关信息。

2. 目录

目录包括设计报告的各章节标题及页码。目录一般按三级标题编写,章节标题的排列建议按以下格式进行:

1.....(第一级)

1.1.....(第二级)

1.1.1.....(第三级)

要求标题层次清晰。目录中标题应与正文中标题一致。

3. 任务书(有专门的格式,直接拷贝即可)

4. 正文

正文是设计报告的详细表达。它占全文的绝大部分,其内容包括方案论证与比较、理论分析与计算、单元电路设计过程、电路测试过程及结果、设计结论等。

(1) 方案论证

方案论证主要论述系统方案的选取及各个方案的比较,从而选出比较好的系统方案,并构建原理框图。

(2) 理论分析与计算

经过比较论证所选方案后,即可对确定的方案进行理论分析,介绍各个电路的工作原理,并对相关电路进行分析设计,对电路的相关参数进行计算,并选取合适的元器件等。

理论计算时,要注意公式的完整性、参数和单位的匹配、计算的正确性,注意计算值和实际选取的元器件参数值的差别。采用仿真软件进行分析时,需要将仿真结果表示出来。

(3) 单元电路的设计

按照总体框图中各个模块实现的功能,对单元电路进行逐一设计并测试,每个模块单独命名;所有单元电路测试结果符合设计要求后,将各单元电路按照信号的流向构建整体电路。

(4) 电路测试过程及结果

整体电路的构建完成后,是否达到设计题目所要求的功能和性能,需要对整体电路进行测试。测试方案要求给出所选的测试仪器、测试方法以及测试数据。测试数据可以用表格、图形或曲线来说明。

(5) 设计结论

整体电路测试后,必须对整体电路进行一个完整的、结论性的评价,测试结果需说明电路通过测试后各个指标是否达到设计题目的各项要求。

5. 附录

在正文中如有较长、不太重要的电路图、表格、图形或曲线等,可放在附录中列出。

6. 参考文献

参考文献应列出在设计过程中参考的主要书籍、刊物、文献、杂志等。

下面给出华北电力大学常用的设计报告的格式。

华北电力大学

课程设计(综合实验)报告

(20 — 20 年度第 学期)

名称: 电子技术综合实验

题目: _____

院系: _____

班级: _____

学号: _____

学生姓名: _____

指导教师: _____

设计周数: _____

成绩: _____

日期: 年 月 日

目 录

《电子技术综合实验》任务书	1
一、综合实验的目的与要求	2
1. 综合实验的目的	2
2. 综合实验的要求	2
3. * * * * * 的设计要求	2
二、设计框图及电路系统概述	3
1. 设计思路	3
2. 系统框图	3
三、各单元电路的设计方案及原理说明、参数计算	4
1. 单元电路 1	4
2. 单元电路 2	4
3. 单元电路 3	4
四、调试过程及结果分析	5
1. 单元电路 1	5
2. 单元电路 2	5
3. 单元电路 3	5
4. 总体电路	5
五、综合实验总结	6
1. 本设计综述	6
2. 收获与体会	6
附录	7
参考文献	8

《电子技术综合实验》任务书(有标准格式,此处省略)

一、设计题目的任务与要求

1. 正文为宋体,五号字 行间距为 21

1.1 -----

1.2 -----

二、设计框图及电路系统概述

1. 正文为宋体,五号字 行间距为 21

1.1 -----

1.2 -----

三、各单元电路的设计方案及原理说明、参数计算

1. 正文为宋体,五号字 行间距为 21

1.1 -----

1.2 -----

四、调试过程及结果分析

1. 正文为宋体,五号字 行间距为 21

1.1 -----

1.2 -----

五、综合实验总结

1. 正文为宋体,五号字 行间距为 21

1.1 -----

1.2 -----

附录(设计框图或流程图、完整的实验原理图、实验所需元器件清单等)

参考文献

[1] 作者 1, 作者 2. 书名. 版本. 出版单位, 出版日期

电子技术基础课程系列教材

电子技术基础	张瑞华
模拟电子技术基础	刘向军
数字电子技术基础	王 贇 刘春颖
电子技术实验指导书	刘向军
模拟电子技术实验指导书	孙淑艳
数字电子技术实验指导书	孙淑艳
电子技术综合实验指导书	孙淑艳

ISBN 978-7-04-043290-9



9 787040 432909 >

定价 22.30 元